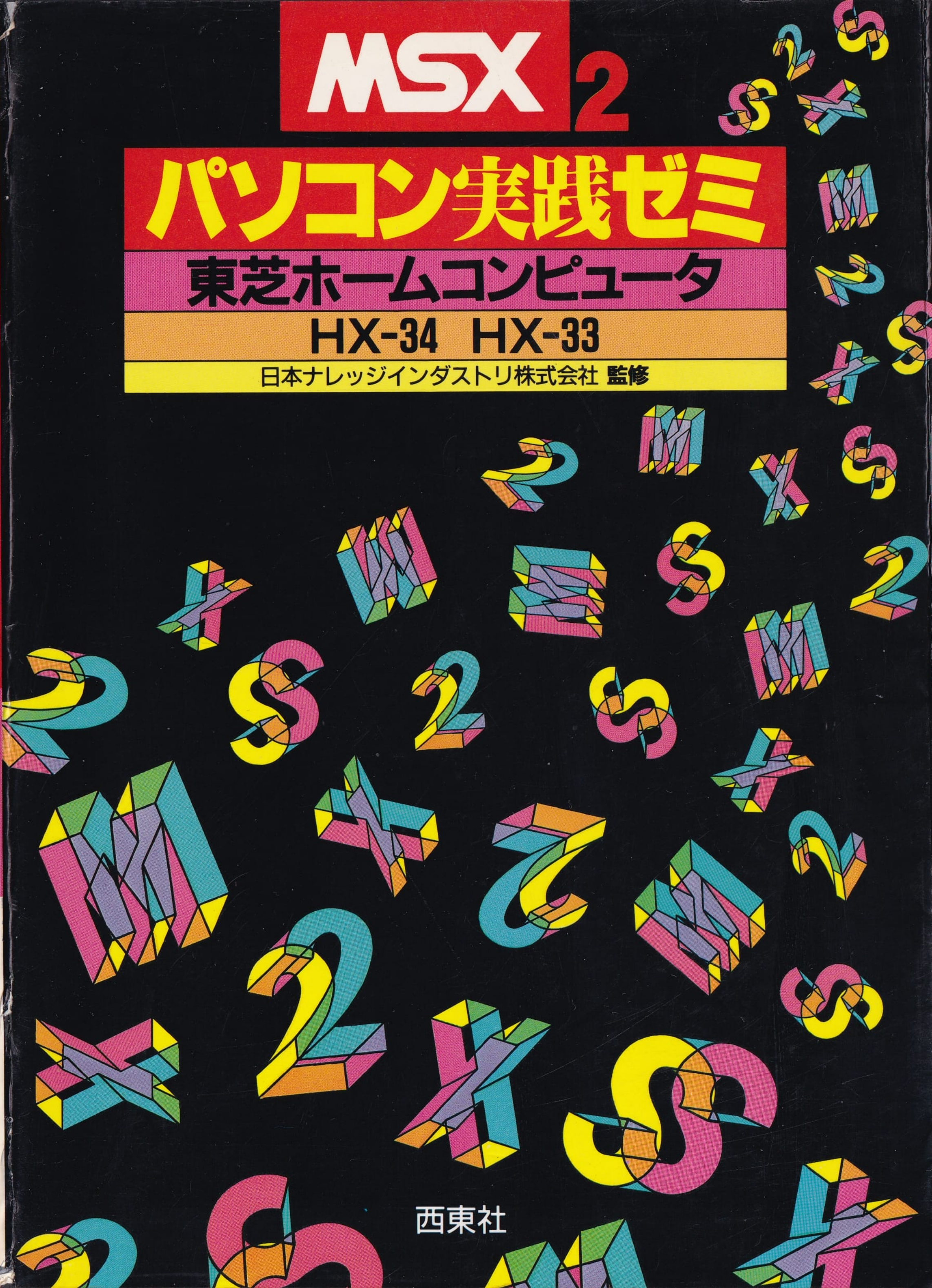
MSX 2 パソコン実践ゼミ

東芝ホームコンピュータ

HX-34 HX-33

日本ナレッジインダストリ株式会社 監修



西東社

MSX 2 パソコン実践ゼミ
東芝 ホームコンピュータ HX-34 HX-33
日本ナレッジインダストリ株式会社 監修
西東社
345

MSX2パソコンはこれでOK!!

ゲームや学習ソフトが手軽に楽しめる入門機から、フロッピーディスク搭載の本格派まで、MSX2パソコンがせいぞろいした。

従来のMSXパソコンに比べ、機能も数段アップして、ユーザーにはたのもしいマシンだ。

しかし、買ってはみたものの〝どうもうまく使いこなせない〟では困りもの。

本書では、MSX2パソコンを100%活用できるよう、ベーシックはもちろんのこと、初歩からわかりやすく解説している。

MSX2パソコンは、これ一冊あればもうOKだ!!

●カバー表4の画面写真
「レイドック」(HX-S253)

MSX2
パソコン実践ゼミ
東芝ホームコンピュータ
HX-34　HX-33

日本ナレッジインダストリ株式会社 監修



西東社

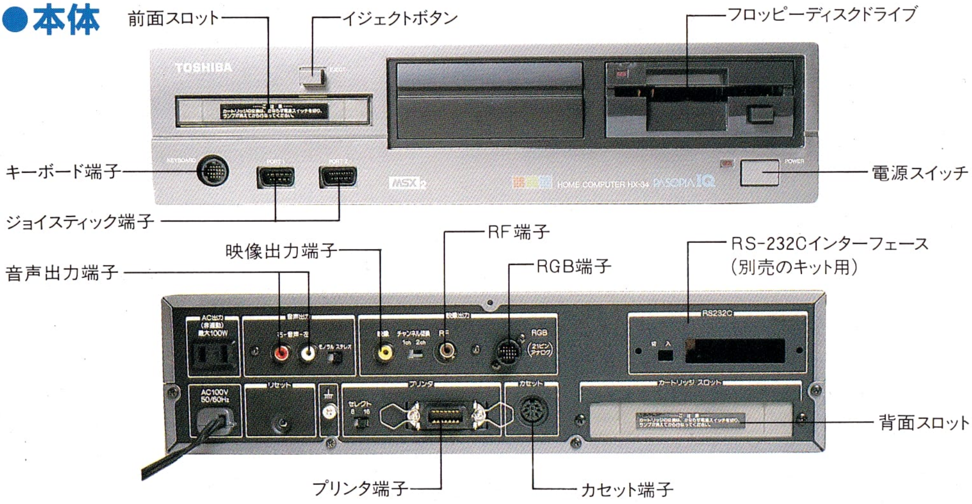
東芝ホームコンピュータ
PASOPIA IQ HX-34 MSX2

プリンタをつなぐだけでワープロになるフロッピーディスク搭載の本格派。

256色同時表示が売り物のMSX2にスーパースターが現れた。日本語ワープロ、英文ワープロ内蔵に加えて、扱いやすい3.5インチ両面ドライブフロッピーディスクを搭載。しかも横置き、縦置きの両方が可能な本体と、テンキーつきキーボードのセパレートタイプ。ホームコンピュータをリードする本格派だ。



●本体



●キーボード



疲れにくいキーボード 打ちやすいキートップ

キーボードは、横長プロポーションのメタリック・タイプ。手前が低く、キートップは本格パソコンではおなじみのシリンドリカル・ステップ・スカルプチャー・タイプ。打ちやすく、疲れにくいという抜群の操作性がウレシイ。



数字の入力に便利な 独立テンキーを採用

家計簿や住所録、売り上げや成績の集計に、なにかと使うことの多いのが数字や計算記号のキー。HX-34では、それがキーボードの右手にまとまった。これで、入力がはるかにラクになるぞ。



3.5インチフロッピーディスク搭載だから、これからのソフトにも対応

記憶容量の大きさと速さ、扱いやすさから、これからの記憶装置やソフトは、マイクロ・フロッピーディスクが中心だ。HX-34なら1基標準搭載で、しかも増設が可能。本格実用ソフト、パソコン通信、ニューメディアへの対応もこれでバッチリだ。

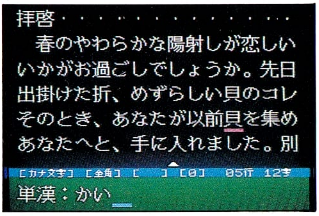




書くから打つへ
日本語ワープロ、英文ワープロを内蔵

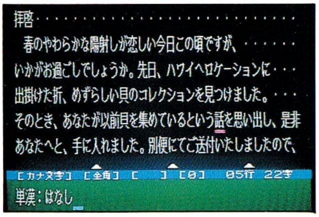
HX−34は日本語ワープロソフトに加えて、漢字ROMを内蔵。プリンタをつなぐだけで、ワープロに変身してしまう。英文ワープロも内蔵で、英文づくりもラクラク。手紙、レポート、案内状から英文レターまで、"打つ"だけでオーケーだ。

単漢入力で即、漢字に



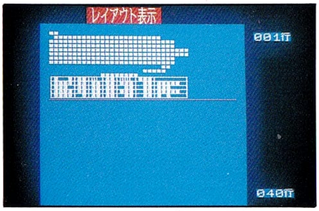
ひらがな、カタカナはそのまま。漢字も読みを入力すれば、キー・ポン！　で即座に漢字に変わる。

15文字か30文字。お好みでどうぞ



文書のスペースは、40文字×169行。ただし、画面に表示されるのは、15文字×6行か、30文字×6行のどちらかを選ぶ。

レイアウト表示で全体を確認



文書の全体を見たり、レイアウトをくふうしたりするときに、この機能が便利だ。画面を見て、簡単にチェックができる。

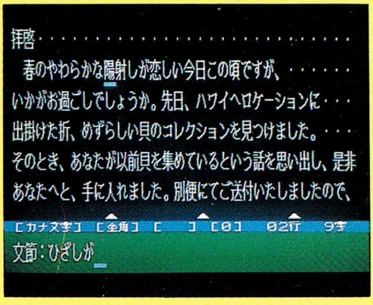
ケイ線もカーソル・キーでラクラク



表づくりや地図、見出しのデザインなど、ケイ線をうまく使えば文書が生きる。引くのも消すのもワンタッチだ。

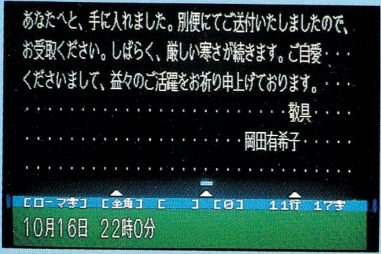
熟語ROMカートリッジで文節変換

熟語ROMカートリッジ（オプション）を使えば、文節単位で漢字変換ができる。たとえば「ひざしが」まで打って RETURN キーを押せば、自動的に「陽射しが」と変換される。収納熟語数は6万7,000。しかも、地名や人名などもそろっているぞ。





■便利な日付、時間入力



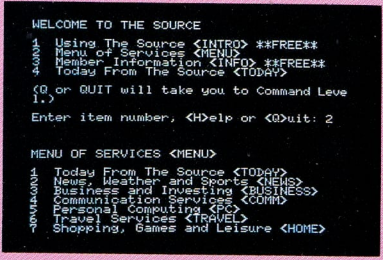
内部時計で、そのときの日付や時間がキーひとつで表示される。思い出づくりにもってこいだ。

■電子メールで通信ネットワーク



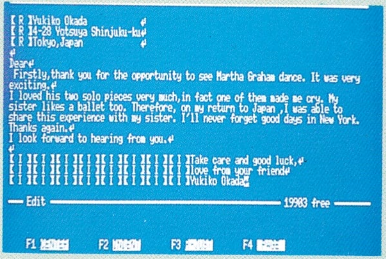
通信機能を使えば、つくった文書を相手のパソコンにリアルタイムで送ることができる。キミだけの通信ネットワークだ。

■海外、国内のデータベースと交信



ニュース、株式市場、イベント情報など、データベースと交信すれば、キミのネットワークはさらに広がる。

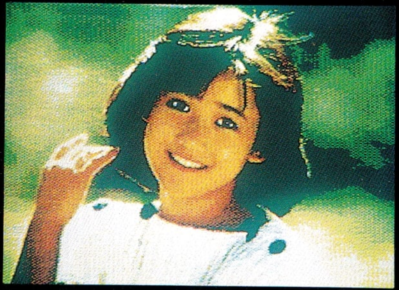
■英文ワープロでペンパルをつくろう



行をそろえる、1単語がとぎれずに収まるなど、便利な機能がいっぱい。英文レターが楽しくなるぞ。

256色同時表示で､リアルなカラーグラフィックスもキミのもの

128Kバイトのグラフィック用のVRAMを搭載。512×212ドットの高解像度、256色同時表示で、リアルな映像づくりが実現した。MSX 2だからこそ、コンピュータ・グラフィックスの世界が広がる。





豊富な周辺機器で



●熱転写プリンタ

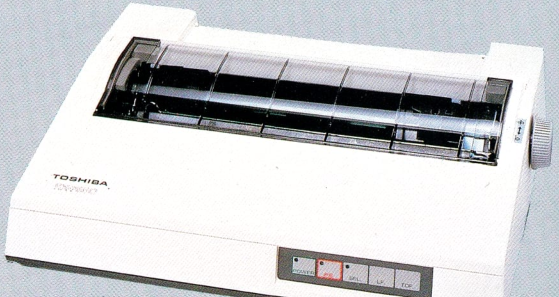
▼リボンカセット



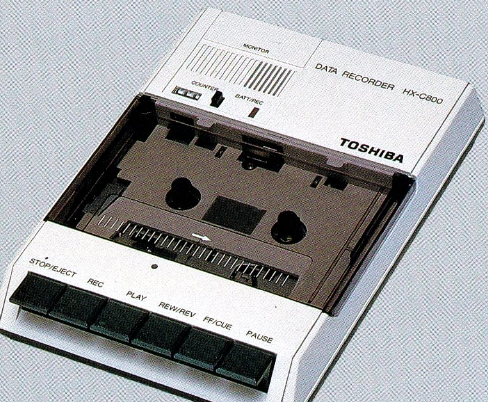
ハガキからＢ４サイズまで、黒、赤、緑、青のうちすきな色で印字ができる。16ドット。音が静かだから、家庭にはうってつけ。



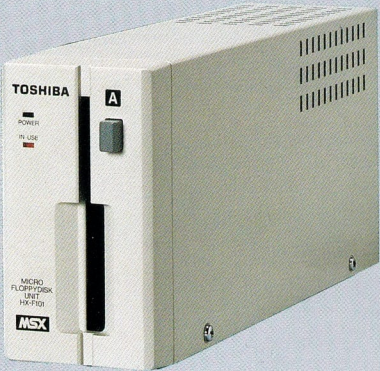
●ジョイスティック
ゲームになくてはならないものがコレ。２トリガーを装備。



●ドットプリンタ
用紙幅４～10インチで、１ページ用紙（単票）への印字も可能。

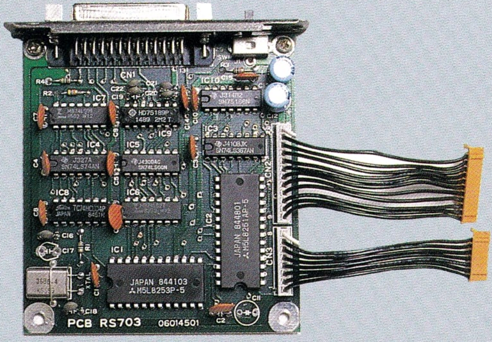


●データレコーダ
データやプログラムの保存に手軽。1200／2400ボーに対応。



●マイクロフロッピーディスクユニット
２DD、１DDの両方のフロッピーディスクが使用可能。MSX-DOSのディスクが付属。

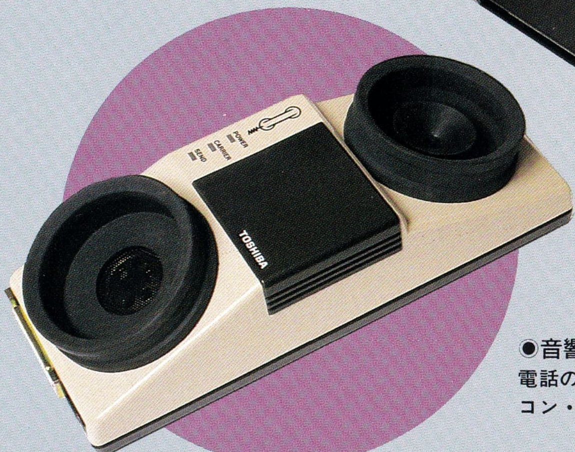
システムアップ





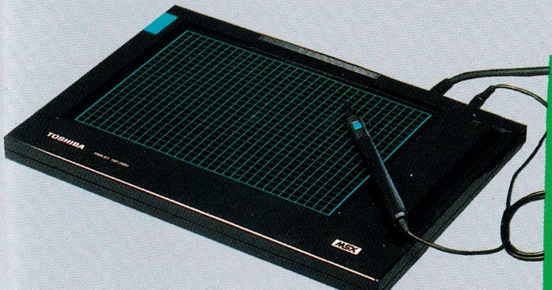
●RS-232Cインターフェースカートリッジ
カートリッジ・タイプだから、スロットに差しこむだけで使える。
(どのMSXにも対応)

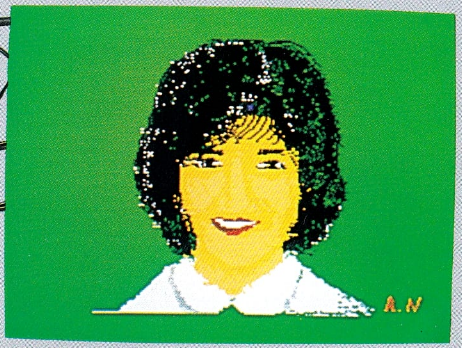
●RS-232Cインターフェースキット
パソコン通信に欠かせない。
本体内に組みこんで使う。



●音響カプラ
電話の受話器をセットして、パソコン・コミュニケーション。

●タブレット





専用のソフトと組み合わせれば、付属のペンでパネル上をなぞるだけで、コンピュータ・グラフィックスができあがる。

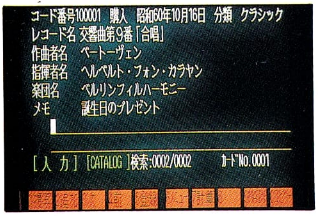
充実ソフトでゲーム&スタディ、家庭をオフィスに

●「はがき君」で年賀状や案内状づくり



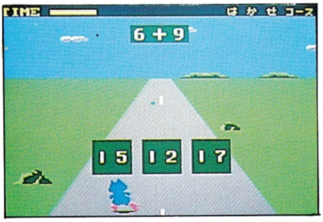
ハガキ印刷用のソフト。謹賀新年や賀正などの定形句、十二支などの定形画が用意されているから、読み出してレイアウトするだけでよい。すきな絵や文字を、自由に描くこともできる。

●「漢字データベース」でレコード、蔵書の整理



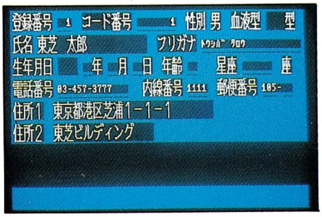
31文字×8行で、約500枚のカードづくりができる。カードの設定はもちろん、検索や印刷も自由自在。レコードライブラリー、蔵書のインデックス……。多彩な用途に対応してくれる。

●すいすいローラー(さんすう)



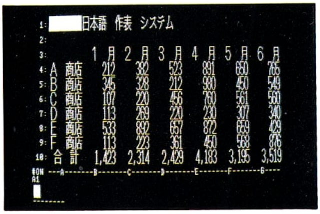
小学校低学年用の四則計算学習ソフト。ローラースケートをはいたネコを動かして、正解のボードに突き進む。ゲーム感覚で楽しみながら覚えてしまう。レベルに合わせてコースが設定されている。

●「宛名書き」で住所録や宛名シールづくり



整理のメンドウな住所録も、これでオーケー。保存件数は約500件で、条件によって特定のグループだけを呼び出すこともできる。シール印刷を使えば、大量の宛名書きもラクラク。

●「漢字スプレッドシート」で集計表づくり



横52×縦128の表の各マス目に、漢字まじりの文字やデータを入力。縦と横の各列や、指定した範囲の合計、平均などが即座に計算できる。検索や並べかえも簡単。

●〈高校入試突破〉英単語完全マスター1200語



必修語、重要語、ハイレベル語が各400語ずつで3コース。学習は、アルファベット順、入試に出る順などを選ぶことができる。学習後のドリルが終わると、ミスしたところだけを反復マスター。

●すぐ、使える/すごく、使える●

パソコンが知的な工具になった

東芝パソピアIQ HX-34 HX-33

気分になったらすぐワープロ

日本語ワープロソフト「漢字君」と漢字ROM(ロム)を内蔵しているから、プリンタをつなぐだけで、すぐにワープロとして使える。手紙でも日記でも、書きたいときに書くのがベスト。キミの頭に浮かんだ名文句を、逃さずワープロしてしまおう。

熟語ROMを使えば、漢字変換も文節単位でオーケー。スピーディに！ スムーズに！キミは、もうワープロ作家だ。

フロッピーでスピーディに保存・読みこみ

3.5インチのフロッピーディスク装置を搭(とう)載(さい)しているから、データやプログラムを保存したり、読みこんだりするのもすばやく完了で、待ち時間のイライラはなし。

プラスチックのケースに入ったカード式だから、気をつかわずに扱えるのもうれしい。

データもプログラムも大量に保存できる。キミのパソコン・ライフも、いよいよ、本格派への仲間入りだ。

画面をそのままキー・ポンで印刷

たとえば、ゲームで最高点を出したとき、あるいは、満足のいくグラフィックスができあがったとき、画面をこのまま残しておきたい。そう、思ったことはないかな。

HX-34、33ならそれが実現する。プリンタの電源を入れて、CTRL と COPY キーを押すだけで、画面がそのまま印刷されるんだ。

パソコン・ワールドが、ますます充実すること、うけあいだ。

軽やかタッチの英文ワープロ

日本語ワープロと同じように内蔵されているのが、英文ワープロ。ちょっとした国際感覚が味わえそうだ。

タイプライターよりも便利な機能を備えているから、単語の切れ目など、細かいことは気にせずドンドン入力できる。

英作文もガゼン楽しくなる。ペンフレンドをつくるのもいい。好きな作家の原文をまねるのもいい。ユニークに使ってほしいな。

256色のリアルなグラフィックス

とにかく、最高512×212ドットの高解像度。加えて、256色の同時表示。最後に、21ピンアナログRGB対応。これ、ベストのグラフィックス・トリオ。デリケートな線もクリアに、色は、あくまであざやかに――。

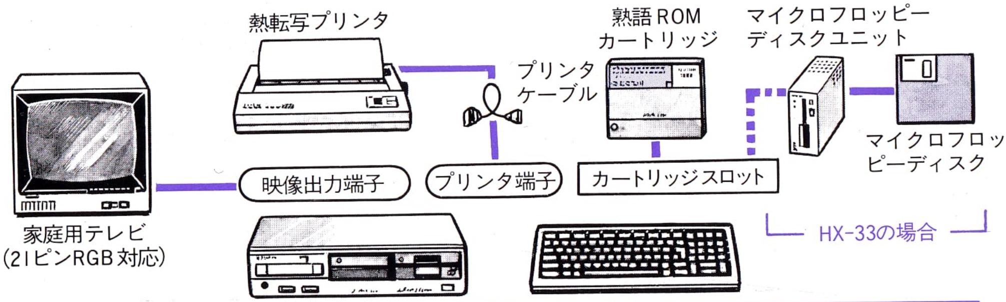
これだけの機能をどこまで生かせるかは、キミのセンスしだい。絵筆の代わりに、ハイテク感覚のキー・タッチで、自在にグラフィックス。ほら、絵心がムズムズしてこないか。

実力派のソフトが勢ぞろい

豊富なソフトで、思いきり楽しめるのが、MSX2パソコンのいいところだ。

それに、すぐに役立つ実用ソフトが加われば、ファミリーでコンピューティング。年賀状や案内状をつくる、住所録をつくる、宛名を印刷する、家計簿をつける、レコードライブラリをつくる、と、パソコンは八面六臂(ぴ)の大活躍。パパやママに、先をこされないようにご用心！

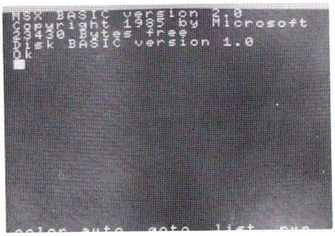
1 パソコンで文章を書こう



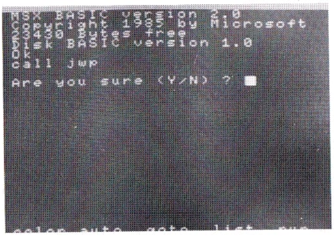
●キー・ポンポンでパソコンがワープロに

日本語ワープロ「漢字君」と漢字ROM(ロム)が内蔵されているから、HX-34、33をワープロに変身させるのに、めんどうな手間はいっさい不要。簡単なキー操作だけでいい。

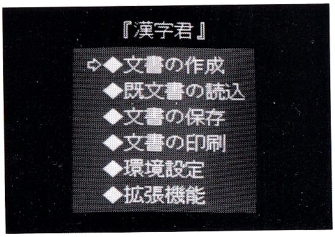
電源を入れて、BASICの初期画面(①)が出たら、「call␣jwp RETURN」(␣は空白のこと)と入力すると、画面には「Are you sure (Y/N)?」と表示される(②)。「日本語ワープロを使いますか」の意味で、YはYes、NはNoの頭文字だ。使うんだったら、Y RETURN と押す。



①初期画面



②(Y/N)?



③メニュー画面

●ワープロには6つの機能がある

さて、画面には、いま、この日本語ワープロでできる6種の機能が表示されているはずだ(③)。これを主メニューという。

◆文書の作成　画面上で文章をつくったり、編集や校正をする。

◆既文書の読込　すでにつくってフロッピーディスクやカセットテープに保存してある文書を、パソコンに読みこむ。

◆文書の保存　つくった文書を、フロッピーディスクやカセットテープに保存する。

◆文書の印刷　つくった文書を、プリンタで印刷する。

◆環境設定　文章作成時に表示する文字の数、印刷に使うプリンタの機種名、文字種(大きさ)、文書の番号などを設定する。

◆拡張機能　熟語の辞書をつくる、用意されていない文字や記号をつくる、文書の送受信をするなど。

主メニューの機能を選ぶには、カーソル移動キーで⇨を動かし、選ぶ機能に⇨を合わせて RETURN だ。

●事前準備をしておけば便利に使える

文章をつくる前に、前準備をしておくことにしよう。主メニューで「環境設定」を選ぶと、設定できる項目が画面に表示される(①)。

それぞれは、次のような内容になっている。

◆表示文字数　画面の1行に表示できる文字数。15文字か30文字かを選べる。つくる文書によって使い分けよう。

◆機種名　接続するプリンタの種類。HX-P560、HX-P550、PA-7253、PA-7270 の4機種の中から選ぶ。

◆文字種　印刷する文字の大きさ。普通(普通の大きさ)、縦倍角(縦に2倍した大きさ)、4倍角（縦横に2倍した大きさ）のうちから選択する。

◆読込み熟語番号　文章中で使った熟語を保存したり、呼び出して使うとき、熟語保存ファイル番号を指定する。

◆読込み外字番号　外字（自分でつくった文字や記号）を保存したり、それを呼び出して使うとき、外字保存ファイルの番号を指定する。

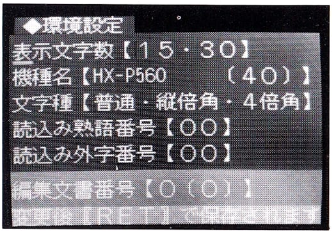
◆編集文書番号　作成、保存、読みこみなどの文書の番号を設定する。

各項目の内容を設定するには、カーソル移動キーで、設定する項目にカーソル(_)を合わせて(②)、表示文字数と文字種は、赤い■記号を→か←のキーで移動させて、選びたいところに合わせる。プリンタの機種は、→キーで内容が変わる(←でもとに戻る)から、選ぶ機種を出す(③)。ほかの項目は、数字を入れる。設定が終われば、RETURN を押す。

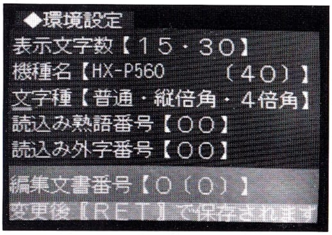
ここでは、表示文字数を15字、機種名をHX-P560として、設定することにしよう。すでに、そうなっているからこのままでいい。あとの項目も、自動的に設定されている値をそのまま使うことにする。だから、RETURN を押すだけですむ。

なお、表示文字数を30文字に設定すると、カーソルを30字に合わせたとたんにメニューの幅が狭くなる(④)。

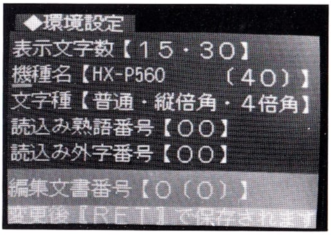
設定した内容は、フロッピーに保存しておくことができるから、いつも同じ条件で使えてつごうがいい。



①環境設定の画面



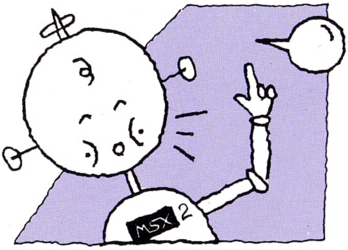
②カーソル移動



③項目の内容を選択



④30文字に設定



●文章の作成にとりかかろう

いよいよ文章をつくる。主メニューで「◆文書の作成」を選べばいいが、その前に主メニューの画面に戻すことが必要だ。STOP キーを押してみよう。画面は主メニューになる。「◆文書の作成」以外の機能から主メニューに戻すには、いつも、この操作が必要だ。覚えておこう。ついでにいうと、「◆文書の作成」から主メニューに戻すには、SELECT を押す。

主メニューが出たら、⇨を「◆文書の作成」に合わせて RETURN 。これが、文章をつくる画面だ(①)。

画面の上５分の４ぐらいは「・」が並んでいる。ここが文章を表示するスペース。「・」１個が１文字ぶんにあたる。１行に15個、縦に６行あるから、この画面は90文字書ける原稿用紙のようなものだ。環境設定で、表示文字数を30字にすると、これが、横30字×縦６行と２倍書ける原稿用紙になる。

左上のすみにある青い「＿」はカーソル。入力した文字はカーソルの位置に表示され、カーソルは右へずれる。

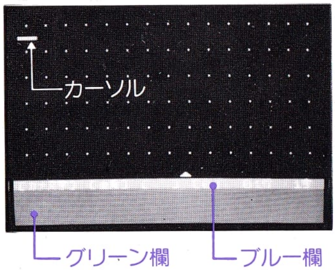
下のブルーの部分は、文字を入力する方法などを表示する部分で、いちばん下のグリーン欄には、漢字やメッセージが表示される。

●ひらがなも漢字もスイスイ書ける

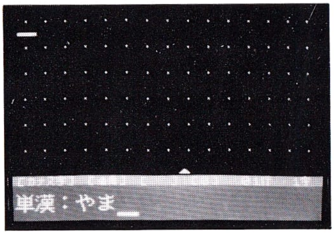
普通の日本語の文章は、漢字とひらがな、カタカナがまじっている。だから、両方を打ちこめる状態にしておくと便利だ。F2 キーを押すと、グリーン欄には「単漢：」という表示が出る。これでオーケー。

キーで入力した文字は、まずグリーン欄に表示される。たとえば、「やま」と入力(②)。SHIFT キーを押しながら RETURN を押すと、「やま」がそのまま原稿用紙の部分に表示される(③)。

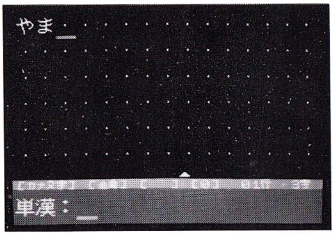
カタカナなら、CAPS キーを押して、横にランプがついている状態にする。あとは、ひらがなと同じ(④)。CAPS キーをもう一度押して、ランプを消すと、ひらがな入力に戻る。



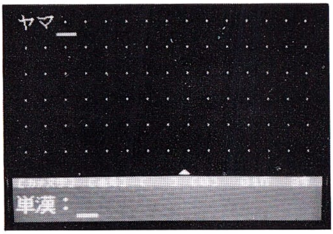
①文書の作成画面



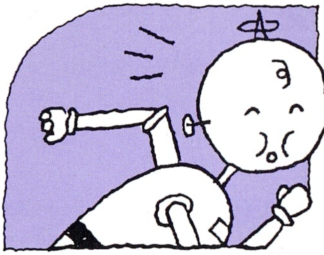
②「やま」と入力



③「やま」が原稿用紙に



④カタカナで入力



●読みの入力で漢字に変換

漢字を使う場合は、漢字１字ごとに読みを入力して、同じ読みのいくつかの漢字の中から選ぶという方法をとる。この方法を単漢字変換という。グリーン欄に表示されている「単漢」は、この意味だ。

たとえば、「山」という字を出してみよう。

まず、ひらがなで読みを入力する。読みは音読みでも訓読み（教育漢字）でもいい。「山」は音読みで「さん」、訓読みで「やま」。ここでは「やま」と入力してみよう（①）。ここまでは、ひらがなの入力と同じだ。

漢字を出すには、RETURN を押す。グリーン欄に「山」と出る（②）から、もう一度 RETURN を押すと、原稿用紙の部分に表示される（③）。同じ読みの漢字がたくさんある場合は、入力した読みの漢字が最大 10 文字表示される。たとえば「さん」と入力して RETURN を押すと、④の画面になる。この10文字に選びたい文字がなかったら、↓を押すと次々に別のグループが出てくる。前のグループに戻すには、↑キーを押せばよい。

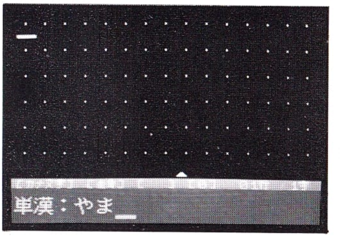
RETURN を押したとたんにブザーが鳴ったら、入力した読みの漢字がないということだ。そのときには、もう一度変換を試みよう。

こうして、目的の漢字が出たら、→か←キーで、カーソルを目的の漢字に合わせる。「山」は初めのグループの２番めにあった（⑤）。これで、RETURN を押すと、「山」が原稿用紙の部分に表示される。

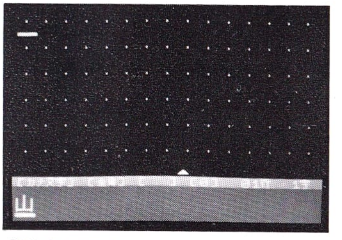
読みの入力をまちがえたら、BS キーを押す。グリーン欄のカーソルのすぐ左の文字が消えていく。消えたら正しい読みを入力して、上と同じ操作をくり返す。

どんな漢字が、どんな読みで出てくるかは、付属の「拡張BASIC　説明書」の漢字一覧表にのっているから、参考にするといい。

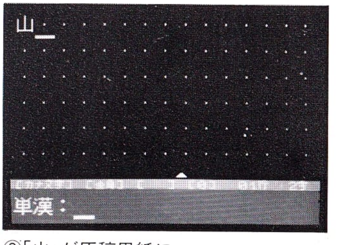
こんなふうにして、日本語の文章がどんどんできあがっていく。画面には、きちんとした楷書体の文字が並んでいて、とても気分がいい。印刷をして、紙に打ち出せば、もっと満足感が高まるにちがいない。



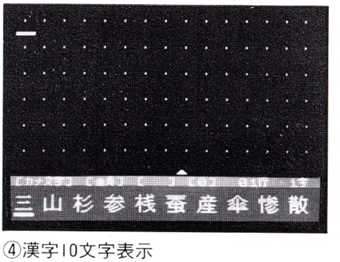
①「やま」と入力



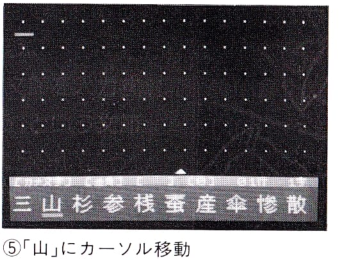
②漢字に変換



③「山」が原稿用紙に



④漢字10文字表示



⑤「山」にカーソル移動

● RETURN で途中で行を変えられる

画面にどんどん入力して、15字を超えるとどうなるか、気をつけて見てみよう。16字めを入力すると、左はしの文字(ここでは空白)が見えなくなる(①→②)。それ以後は１字入力するたびに、左はしの文字が消えていく。

それでも、入力を続けていくと、41字めを入力したとたんに、画面の表示が右に移動した感じになって、最初に入力した部分があらわれてくる。

つまり、画面には、15字しか表示されないが、実は40字ぶん入力できるということなのだ。これは、表示文字数を30字に設定しても同じこと。それに、行も画面の6行ではなく、169行ぶんが入力できる。

１行の文字数を40字以内におさえたいときは、行を変えるところまできたら、次のようにする。

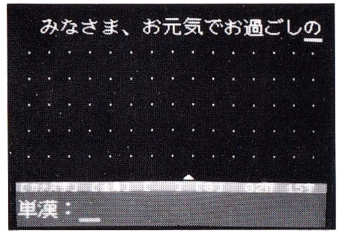
❶グリーン欄に何も表示されてない状態にする

読みが入力されていれば BS キーを押し続けて消し、次に STOP キーを押して「単漢：」を消す(③)。

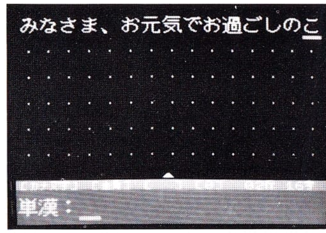
❷ RETURN を押す

カーソルが次の行の先頭に移る(④)。

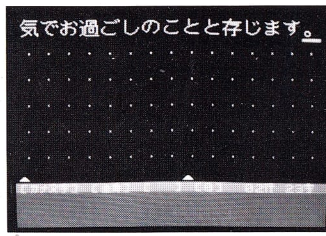
ところで、「漢字君」の機能には、グリーン欄に何も表示されていない状態でしか使えないものがある（本書では★印をつけてある)。その場合には、上記の操作で表示を消してから、その機能を使うこと。



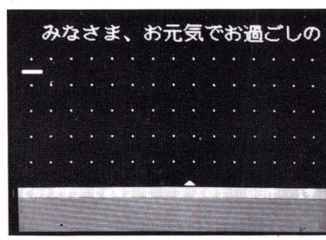
①15字を入力



②16字め(左の１字が消える)



③グリーン欄の表示を消す



④カーソルが移動

●カーソルを動かすには2通りの方法がある

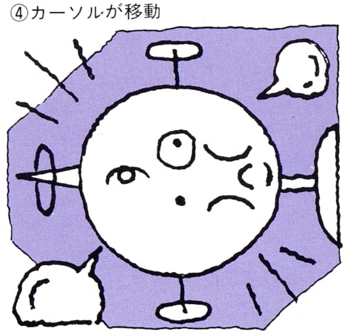
カーソルを動かして原稿用紙の自由な位置に文字を入力するには、グリーン欄に表示があるかないかによって、操作が違う。

■グリーン欄に表示がある場合

SHIFT を押しながら、カーソル移動キー。 SHIFT を押さないと、グリーン欄のカーソルが動くだけだ。

■グリーン欄に表示がない場合

カーソル移動キーだけでいい。 ESC を押しながら ↑ を押すと、カーソルがどの位置にあっても文章の先頭、つまり１行の１字めにカーソルが移動する。



●よく使う熟語や短文は登録して使おう

たとえば、「案内」ということばを入力する場合、案と内を別々に入力するより、「あんない」で一度に変換できたら便利だ。熟語登録機能（★）を使うと、それができる。「案内」を「あんない」という読みで登録しておくと、その読みで漢字に変換できるんだ。

ちょっと使ってみよう。

グリーン欄の表示を消して、登録したい熟語の最初の文字の下にカーソルを合わせたあと、 SHIFT を押しながら F3 キーを押す。すると、グリーン欄には「熟語登録：どこまで？」と表示される(①)。

次に、→を押すと赤いカーソルが出て右へ動くから、登録したい熟語の最後の文字に合わせて RETURN 。グリーン欄には「読み：」と表示されるので、熟語の読み(略称でもよい)を入力しよう(②)。これで、 RETURN を押せば、登録完了だ。

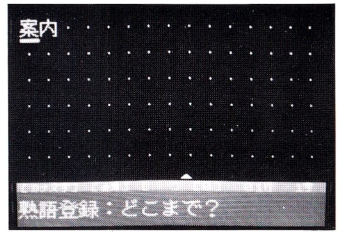
熟語と同じように、短い文章を登録しておくこともできる(★)。 F3 の代わりに F4 のキーを使い、読みの入力がいらないこと以外は、操作は熟語登録と同じだ。

●登録した熟語を使ってみよう

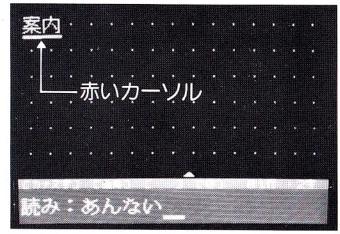
登録した熟語を使うには F3 キーを押す。これでグリーン欄に「熟語：」と表示されるから、読みを入力して(③) RETURN を押せば、グリーン欄に熟語が表示される(④)。もう一度 RETURN を押せば、グリーン欄の熟語が文章作成の部分に移る(⑤)。

15字まではひとつの熟語として登録できるから、「東京都消費者センター新宿支所」などという長い字も一度で変換して使える。いらなくなったものは削除できるし、拡張機能の「熟語辞書作成」でフロッピーディスク（以下、フロッピー）に保存しておくこともできる。

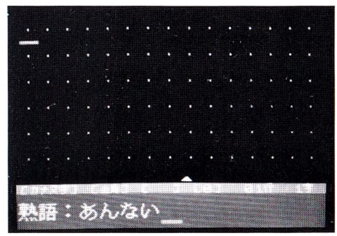
使いながら「これは」と思う単語は、すぐ登録しておくといい。限度は、熟語の文字数と読みの文字数の合計で、約6000字まで。登録した熟語が多くなるほど、変換がスムーズにできるようになる。



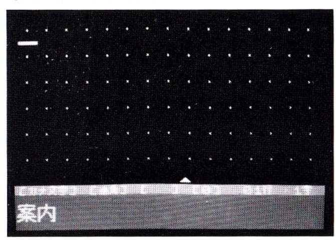
①熟語登録の画面



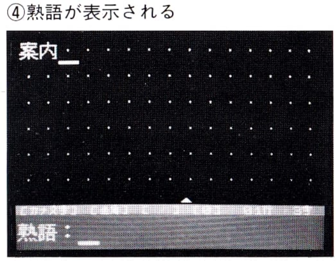
②読みを入力



③熟語の呼び出し



④熟語が表示される



⑤熟語が原稿用紙に

●熟語ROMを利用すると文節ごとに変換可能

別売の熟語ROMカートリッジ（HX-M210）を使うと、単漢字変換や熟語登録機能を使うよりも、もっと速く文章をつくれる。文節単位で簡単に漢字に変換できるようになるからだ。

文節とは、ことばを発音するとき、意味がわかり、不自然ではないように区切った最小の単位のこと。間にネということばをはさんでみるとわかりやすい。「私の好きな教科は理科です。」という文章だと、「私のネ　好きなネ　教科はネ　理科です。」と、４つの文節になる。

文節単位で変換できると、たとえば「私は」を入力する場合、「私」を変換後に「は」を入力するという手順をふまなくても、「わたしは」を１回で変換できる。キー操作がそれだけ少なくてすむ。「わたしは、」というふうに、読点（。）や句点（、）があってもかまわない。

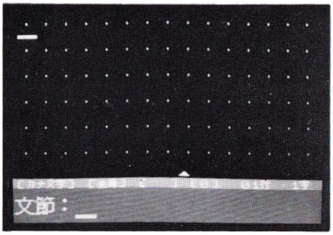
読んだとおりの読みを入力すればいいから、単漢字変換よりも、ずっと自然に文章をつくれる。この方法を熟語変換という。

ちょっと試してみよう。熟語ROMをセットしたら、まず F1 キーを押す。するとグリーン欄に「文節：」という表示が出る（①）。これで、文節変換ができる状態になった。

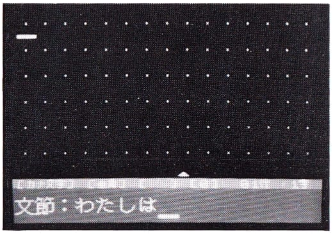
あとは文節の読みを入力して（②）、RETURN を押すとグリーン欄に漢字まじりの文節が表示される（③）。これが第一候補で、↓ キーを押せば次の候補が表示されていく。目的の文節になったら、RETURN を押すと、文章作成部分に文節が移る（④）。

この熟語ROMに入っている熟語は6万7000もある。これだけあれば、たいていの文章はまにあう。

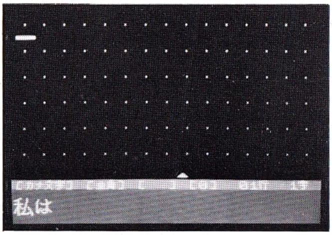
それに、一度選択した熟語は、次に同じ文節を入力すると、第一候補になる。たとえば、「いこう」という文節は、最初、「意向→以降→移行→憩う→行こう→いこう」の順で変換されるが、一度「行こう」を選択すると、次に変換するときには「行こう」が最初に出てくるというわけ。パソコンが覚えておいてくれるんだ。



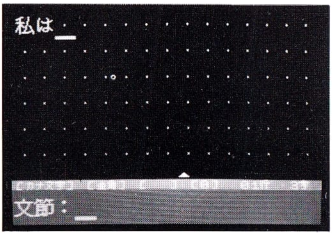
①「文節：」の表示



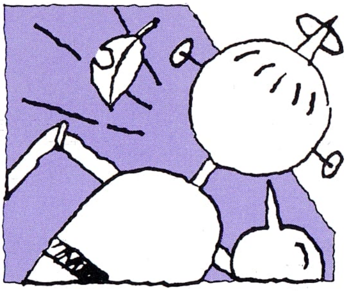
②読みを入力



③文節が変換



④文節が原稿用紙に



●のってきたら読みだけ入力、あとで変換

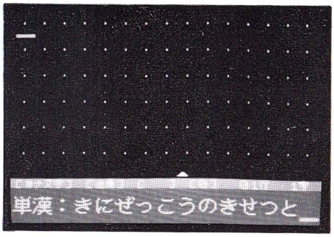
書きたい文章がどんどん頭に浮かんできて、いちいち変換するのがめんどうなときがある。そんなときに試してほしいのが、一括(いっかつ)入力という方法だ。

グリーン欄には、最大984文字の読みが入力できるようになっている。これを利用して、とにかく区切りのいいところまで、ひらがな（読み）で入力してしまう。この間は、漢字に変換する操作をしない。これなら、グンと入力スピードがあがるだろう。

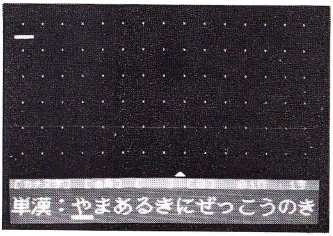
実際にやってみるとわかるけど、グリーン欄の文字は12字以上の読みを入力すると左へスクロールして、左端は消えていく(①)。といっても、ほんとうに消えてしまうわけではなく、画面で消えるだけだから心配はいらない。入力がひと区切りついたら、↑キーを押して、入力した文章の先頭を出し(②)、それから、おもむろに漢字に変換していく。

変換する文字のすぐ後にカーソルを移動して RETURN を押す、という作業をくり返すわけだ(③)。ひらがなのままなら SHIFT を押しながら RETURN を押す。

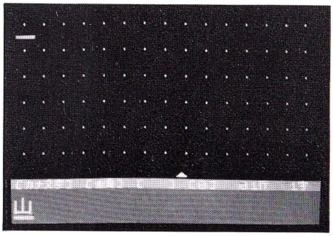
ただし、単漢字変換の場合は、漢字は1字ずつの読みで入力すること。熟語変換なら、こんな注意は不要だ。



①12文字以上を入力



②先頭の呼び出し



③漢字変換

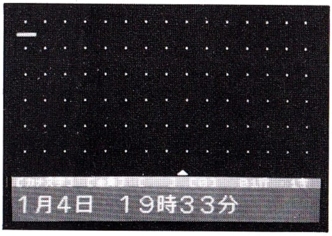
●日時の表示もキー操作だけでできる

本体の内部では時計が動いていて、日付や時刻がすぐにわかるようになってる。これを利用して、文章中に、作成した月日や時刻を入れることができる。

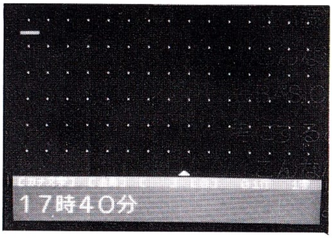
単漢字変換の状態で、「にちじ」と読みを入れ RETURN を押すと、「○○月○○日　○○時○○分」と、そのときの日時に変換される(④)。これで、RETURN を押すと、この数字が文章作成の部分に移る。

ほかにも、「ひにち」または「ひづけ」で「○○月○○日」、「じかん」または「じこく」で「○○時○○分」と変換される(⑤)。

手紙の最後に書いたり、作成した文章の覚書にしたりして利用しよう。



④「にちじ」変換



⑤「じかん」変換

●何回訂正してもしあがりはスッキリ

つくった文章は、次のようにして、何度でも訂正できる。

◎**重ね書き**　もとの文字数と訂正する文字数が同じならば、上から重ねて書く。もとの文字は消えて、新しい文字に変わる。

◎**削除**　よけいな文字を消すには、グリーン欄の表示を消してから削除する文字の右側にカーソルを合わせて、BSキーを押せばいい(①)。削除する文字数が多い場合は、削除する文章の先頭にカーソルを合わせ、ESCを押してからDELキーを押し、削除する範囲を指定してRETURNという方法が便利だ。

◎**挿入**　グリーン欄の表示を消してから、文字を挿入したい位置にカーソルを合わせて、INSキーを押す。カーソルは紫色になって、挿入できる状態になる(②)。あとは、挿入する文字を入力すればいい。挿入が終わったら、もう一度INSキーを押す。カーソルはもとの青色に戻り、挿入状態が解除される。

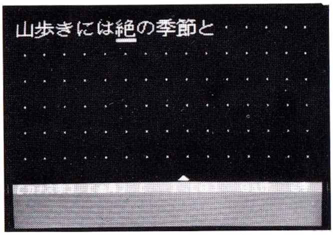
手書きで原稿を書くと、訂正を重ねるほど原稿がきたなくなるが、ワープロなら、そんな心配はまったくない。訂正のあとのない、きれいな文書ができあがる。

●いろんな機能はメニューで選べる

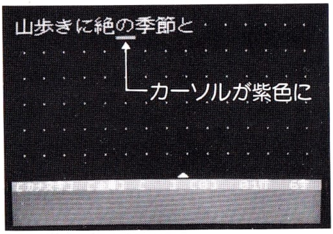
「漢字君」は、キミのセンスに合った文章をつくれるよう、豊富な機能を持っている。機能を使うのはキー操作でもできるけど、機能が一覧できるメニューを表示して、そこから選ぶようにすると便利だ。キー操作を忘れたときにも助けてくれる。

このメニュー表示機能（★）は、グリーン欄を消してから、ESCを2回押す。これで、機能のメニューが表示される(③)。

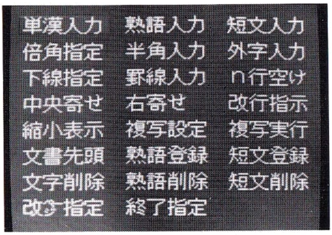
この中から、機能を選ぶには、カーソル移動キーで赤い▬を目的の機能に合わせて(④)、RETURNを押せばいい。試しに、文字を横に2倍の大きさにする「倍角指定」を選んでみた(⑤)。



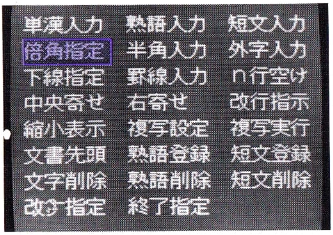
①削除（「は」を削除）



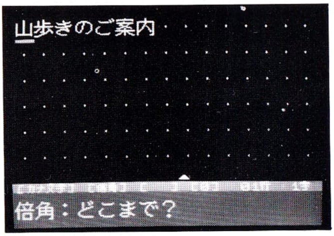
②挿入



③メニュー表示



④倍角指定を選ぶ



⑤倍角の範囲指定の画面

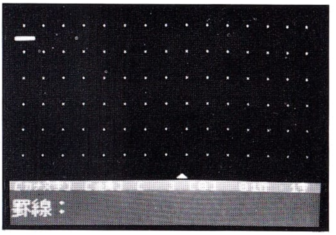
●ケイ線もカーソル移動キーでスイスイ書ける

表をつくったり、簡単な地図を描いたり、見出しを囲んで強調したり、といろいろと便利なのがケイ線機能だ(★)。カーソル移動キーで自由自在に線を引けるから、大いに利用しよう。

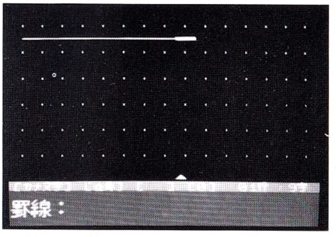
ケイ線を引くには、まず、グリーン欄を消してから、ESC、5と押すか、前ページのメニュー画面で選ぶ。すると、グリーン欄に「罫線：」と表示される(①)。これで、あとは、GRAPHキーを押しながら、カーソル移動キーで線を引きたい方向にカーソルを動かせばいい。カーソルが移動したあとがどんどん線に変わる(②)。

縦線と横線がぶつかる角も、きれいに書ける(③)。線を消したいときは、SHIFTを押しながらカーソル移動キーを使う。これも、カーソルが移動したところの線が消えていくから手間がない。

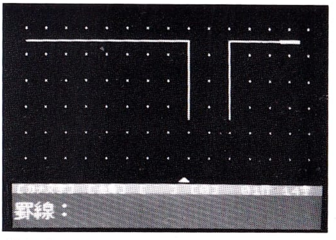
キー操作が簡単だから、地図のように、ちょっと複雑なものでも手早くつくれるぞ(④)。



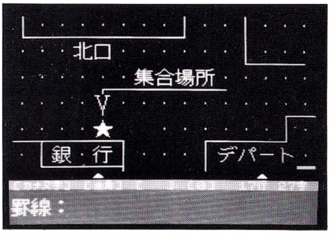
①罫線機能の画面



②線を引く



③角もきれいに



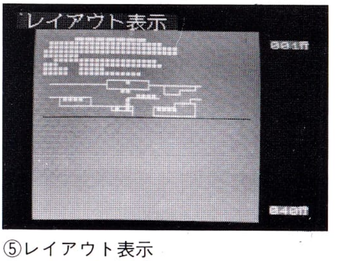
④地図の作成

●文章のレイアウトを確認しながら

長い文章をつくる場合など、ちょっと困るのは、画面には一部しか表示されないことだ。これでは、全体の感じがつかめない。見栄えのいい文書をつくるには、全体がどんなぐあいか、文字と地図のバランスはどうかなど、ときどきチェックしながら進めるほうがいい。

レイアウト機能（★）を十分に利用して、作成の途中や、いちおう、しあがった段階でレイアウトを検討しよう。グリーン欄の表示を消して、ESC、8。メニュー画面の「縮小表示」を選んでも同じだ。これで、カーソル位置の上下20行ずつ、全部で40行（1行は40字）ぶんのレイアウトが表示される(⑤)。↑か↓のキーを1回押すごとに、20行ぶんずつ移動するから、ほかの部分のレイアウトを見ることもできる。

もとの文章作成画面に戻すには、STOPキーを押せばいい。修正と確認をくり返して、気にいったレイアウトにしよう。



⑤レイアウト表示

●キー操作で機能を選べれば一人前？

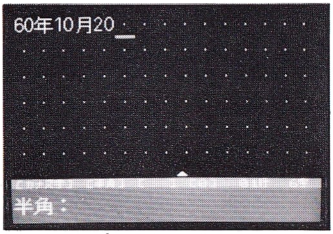
初めは、いちいち機能のメニュー画面を出して使うにしても、慣れてくると、キー操作が少なくてすむような方法をとりたくなるもの。これまで説明しなかった機能も含めて、どういうキー操作で、どの機能が使えるようになるのか、一覧表にしてみた。覚えるまでは、すぐ見える場所にはっておくといい。

■ファンクション・キーを使う

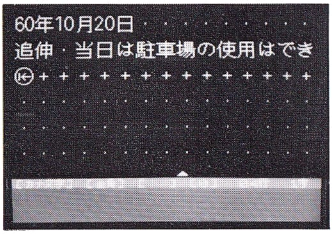
	F1	文節ごとの入力（文節変換）
	F2	漢字1字ごとの入力（単漢字変換）
	F3	登録した熟語の入力
	F4	登録した短文の入力
	F5	文字を半分の大きさにする（半角①）
SHIFT を押しながら	F1	ページを変える（改ページ）
	F2	終わりの記号を入れる（ENDマーク②）
	F3	熟語を登録する（熟語登録）
	F4	短文を登録する（短文登録）
	F5	文字を倍の大きさにする（倍角③）

■ ESC キーを使う

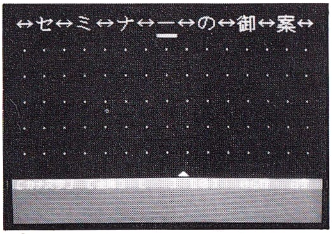
ESC を押してから	0	短文を削除する
	1	文字を中央に移動（センタリング）
	2	文字を右端に寄せる（右寄せ④）
	3	行を変える（改行）
	4	アンダーラインを引く（下線⑤）
	5	線を引く（ケイ線）
	6	複写部分の登録（コピーの登録）
	7	登録部分の複写（コピーの実行）
	8	レイアウトを表示する
	9	登録した熟語を削除する
	—	外字一覧表を表示する
	↑	文章の先頭を呼び出す
	INS	文章の途中で行をあける（行空け）
	ESC	機能メニューを表示する



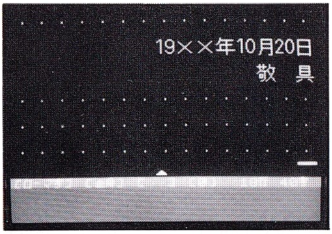
①半角文字



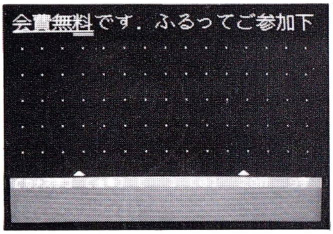
②ENDマーク



③倍角文字



④右寄せ



⑤下線

●プリンタを接続して印刷する

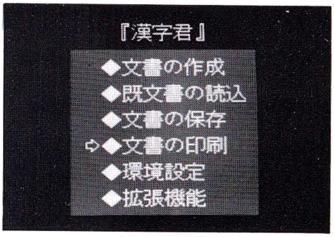
文章ができたら、いよいよ印刷にかかる。

まず、プリンタが正しく接続されているかどうかを確認しよう。接続できていれば、電源を入れる。もちろん、漢字が印刷できるし、B5～B4の大きさまで対応。ハガキに印刷することもできる。HX-P560は熱転写方式だから、音が静かなのもいい。自分のつくった文章が、きれいな文字で軽やかに印刷されてくるようすは、ちょっとワクワクものだ。

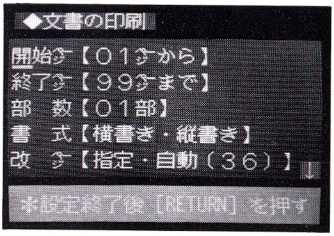
用紙をセットして、プリンタの準備はこれでよし。

文章の最後にENDマークを入れたら(これがないと、印刷に時間がかかる)、SELECT キーを押して主メニューを出し、「◆文書の印刷」にカーソルを合わせる(①)。RETURN を押すと、印刷メニューが表示される(②)。

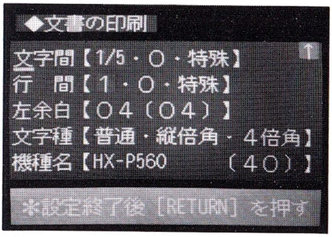
メニューは全部で2画面あって、カーソルを押しつづけていると、次の画面が表示される(③)。開始ページや終了ページ、部数、書式(横書き、縦書きの別)、文字種(文字の大きさ)などのほか、文字間や行間、余白のとり方なども指定できる。よく検討して、文章の内容に合ったかたちで印刷しよう。



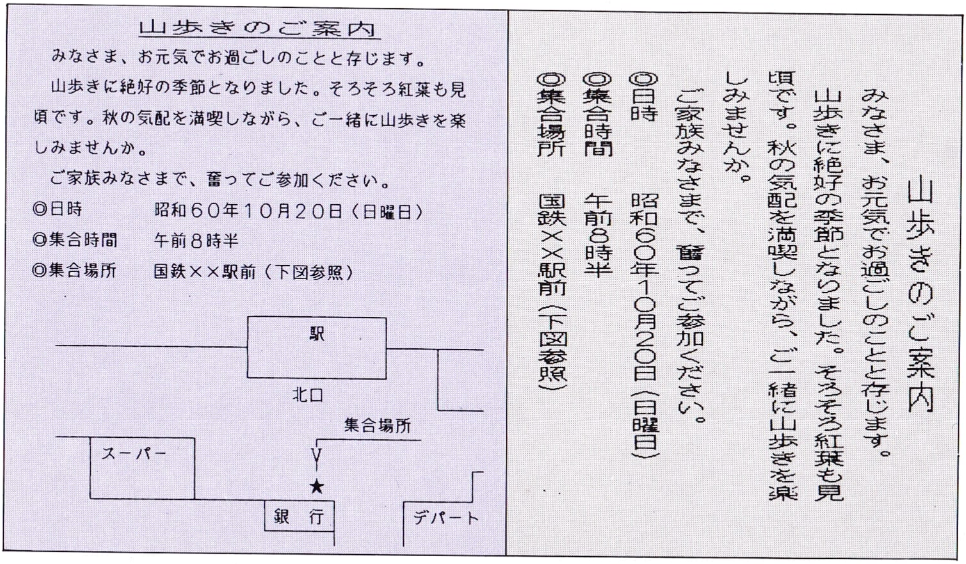
①文書の印刷



②印刷メニュー



③次のメニュー画面


山歩きのご案内

みなさま、お元気でお過ごしのことと存じます。

山歩きに絶好の季節となりました。そろそろ紅葉も見頃です。秋の気配を満喫しながら、ご一緒に山歩きを楽しみませんか。

ご家族みなさまで、奮ってご参加ください。

◎日時　昭和６０年１０月２０日（日曜日）
◎集合時間　午前８時半
◎集合場所　国鉄××駅前（下図参照）

山歩きのご案内

みなさま、お元気でお過ごしのことと存じます。

山歩きに絶好の季節となりました。そろそろ紅葉も見頃です。秋の気配を満喫しながら、ご一緒に山歩きを楽しみませんか。

ご家族みなさまで、奮ってご参加ください。

◎日時　昭和６０年１０月２０日（日曜日）
◎集合時間　午前８時半
◎集合場所　国鉄××駅前（下図参照）

●つくった文書はフロッピーに保存

印刷したあと、つくった文書は、フロッピーに保存するといい。記録しておけるし、同じような文章を書くときに、少し手直しすれば使えるからだ。

保存用のフロッピーをセットしたら、「◆文書の印刷」の状態から STOP キーを押して主メニューを出し、「◆文書の保存」に⇨を合わせて RETURN 。さらに、「⇨◆文書の保存」で RETURN を押す。画面の下に名前を入力する欄が表示されるから、6文字以内（漢字は不可）で、保存する文書に名前をつけよう。内容がすぐわかるような名前がいい。たとえば、「やまあるき」とする（①）。

これで、 RETURN を押せば保存が始まる。画面右上には「保存中」という表示が出ているはずだ（②）。この表示が消えれば、保存終了。フロッピーだからスピーディに保存が終わる。

正しく保存されたかどうかは「◆文書の照合」で確かめることができる。

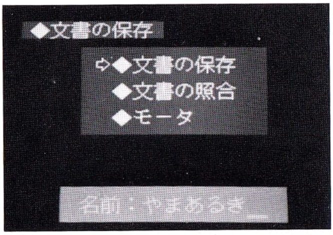
1枚のフロッピーにはA4で約560ページぶん（2DDという表記のあるもの）が保存できる。本の2冊ぶんはゆうゆうと入ってしまうぞ。

●希望の文書をすぐに呼び出せる

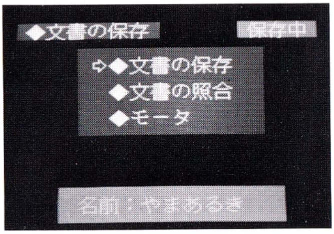
フロッピーに保存してある文書は、必要なときに、すぐに呼び出せる。

まず、目的の文書が保存してあるフロッピーをセットしてから、主メニューで「◆既文書の読込」を選んで、さらに「◆既文書の読込」を選ぶ。すると、約1秒で、保存文書の一覧表が出てくる（③）。目的の文書に赤い■を合わせて RETURN を押すと、画面の右上に「読込中」と表示が出る（④）。読みこみが終わると、この表示が消える。

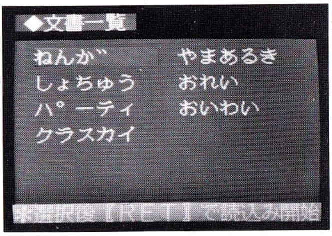
読みこんだ文書を画面に出すには、 STOP キーを押して主メニューに戻し、「◆文書の作成」に⇨を合わせて RETURN 。画面に表示された文書は、訂正や削除などで自由に加工できる。



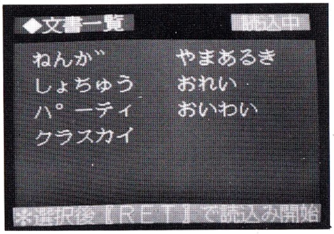
①文書名を登録



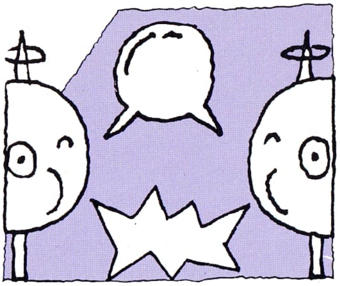
②保存中



③保存文書リスト



④保存文書の読みこみ



●ワープロ通信で電子レターをやりとりすれば

「漢字君」には、通信機能がついているから、同じ機種どうしなら、電話回線を利用して文書をやりとりできる。とくに改まった文章でなくても、交互に送信と受信をくり返して会話を楽しむこともできる。受信した文書はフロッピーに保存したり、プリンタで印刷したりすることもできる。電話やふつうの手紙とは、ひと味違ったコミュニケーションができるぞ。

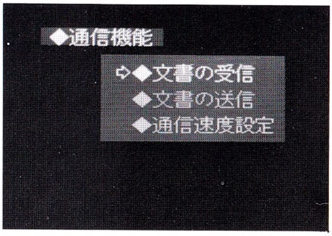
ワープロ通信には、RS-232C インターフェースキットと音響カプラという装置が必要だ。両方の装置と電話の受話器を正しくセットしたら、通信の準備を始めることにしよう。

主メニューで「◆拡張機能」を選んで、そのメニューが出たら、さらに「◆通信機能」を選ぶ(①)。このメニューで⇨を「◆文書の送信」に合わせて RETURN 。これで、文書を送れる状態になる(②)。相手のパソコンが受信する状態になっていれば、RETURN で文書が送られていく。文書を送っている間は、画面の右上に「送信中」と表示され、送っている文章が画面下に流れるように表示されていく(③)。まさに、送っている感じだ。

受信する場合は、「◆文書の受信」に⇨を合わせて RETURN 。さらに、追加受信か新規受信を選んで RETURN 。これで受信が始まり、画面右上には「受信中」の表示が出る(④)。この表示が消えれば、受信終了だ。

ところで、パソコン通信では、送信側と受信側で、情報をやりとりする速度が同じであることが条件だ。この速度をボーといい、初めは300ボーになっている。音響カプラの場合は300ボーに決められているから、相手が300ボーに合わせていれば、特別な操作をしなくても、このままでいい。

もし、もっと速くやりとりしたければ、通信速度を変えることができる。通信機能のメニュー(①)で、「◆通信速度設定」を選ぶと、5段階で選ぶことができる。数字が大きいほど速くなるから、4800ボーを選べば最高速になる。⇨を合わせて(⑤)、RETURN で設定できる。



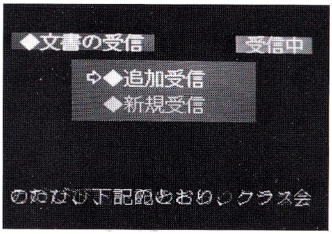
①通信機能の選択



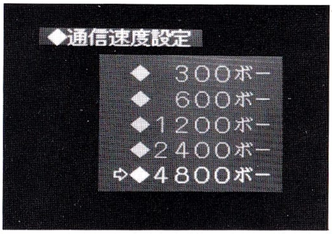
②文書の送信準備



③送信中の画面

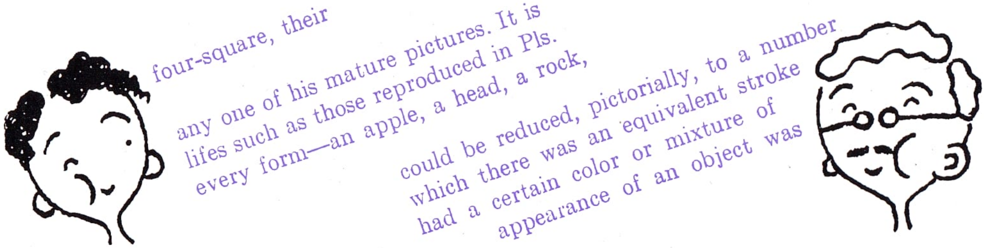


④受信中の画面



⑤通信速度の選択

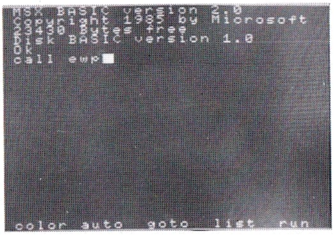
2 英文ワープロでヘミングウェイ



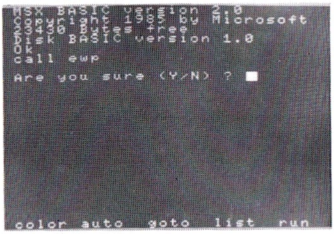
●英文ワープロにも即座に変身

HX-34、33には、日本語ワープロとともに英文ワープロも内蔵されている。外国のペンフレンドに手紙を書く、英語で日記を書く、友人と英文レターをやりとりするなど、いろいろに使える。もちろん、英作文など英語の勉強に使うのもいい。アメリカ人がタイプライターを打っている気分で英語の文章が書けるから、勉強もはかどるはずだ。

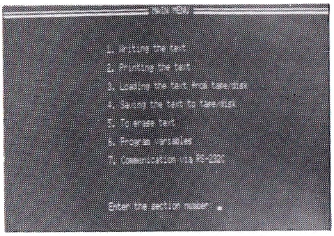
英文ワープロを呼び出すには、BASICの初期画面で「call␣ewp」と入力して(①)、RETURN を押すと、「Are you sure (Y/N) ?」という表示が出る(②)。意味は、日本語ワープロのときと同じだから、Y RETURN と押せばいい。これで、パソコンは英文ワープロになった。システムも、日本語ワープロと同じだ。



①英文ワープロの呼び出し



②(Y/N)?



③主メニュー

●英文ワープロに7つの機能

上のようにキー操作をすると、画面には主メニューが出たはずだ(③)。この英文ワープロには、これだけの機能がある。それぞれの内容を簡単に説明しておこう。

1. Writing the text　文章の作成、編集、校正を行なう。
2. Printing the text　文書をプリンタで印刷する。
3. Loading the text from tape/disk　テープやフロッピーに保存してある文書を本体に読みこむ。
4. Saving the text to tape/disk　作成した文書をテープやフロッピーに保存する。
5. To erase text　作成した文書を消去する。
6. Program variables　画面の背景色や印刷書式を設定。
7. Communication via RS-232C　文書の通信を行なう。

メニューの選択は、1～7の数字で選定して RETURN を押す。

また、どれかの機能を実行してる状態から、主メニューに戻すには ESC キーを押しつづければよい。

●英語で文章を書いてみよう

文章を書くには、まず「1. Writing the text」を選んで、文章を作成する画面にする。1 RETURN で、その画面になる(①)。あとは、どんどん入力していこう。画面には、1行に77文字（指定により79文字）で16行入る。英文で気になるのは、ひとつの単語が2行にまたがってしまうことだが、このワープロは、それをうまく処理してくれる。

実際に試してみよう。画面の右端にsisterという単語がきた(②)。このままなら、siとsterがバラバラになってしまう。だが、実際には、sisterは自動的にひとつの単語としてまとまって次の行に移る(③)。これをワードラップ機能という。この機能のおかげで、単語の切れ目を気にせずに入力できる、というわけだ。

この機能によって、行の最後には数文字ぶんの空きができるが、印刷するときには、ジャスティフィケーション機能によって、行の単語が適当に間隔を広げて、空いたスペースを埋めてくれる。

だから、印刷したものは、単語が2行にまたがっていたり、文字列の右端がデコボコしたりすることはないわけだ。

●多彩な機能で編集もラクラク

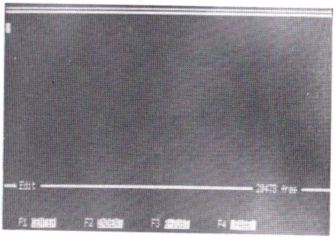
削除、挿入、センタリング、右寄せ、複写、通信など日本語ワープロと同じような機能以外にも、英文ワープロに独特な機能もある。

◆Find機能　文章中から指定した単語を探し出す。固有名詞や特定のことばを確かめるのに便利。

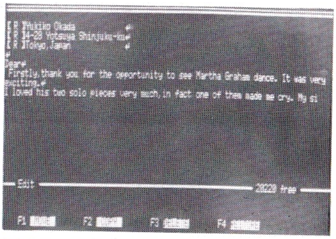
◆Replace機能　ある文章や単語を、別の文章や単語に自動的に置きかえる。文字数が違っても、それに合わせて全体の送りや詰めが行なわれる。

◆Move機能　ある文章や単語を、別の位置へ移動。文章の一部を入れかえたり、構成を変えたりするのに便利。

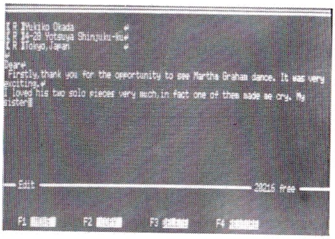
◆ブロック編集　文章の一部をまとめて、移動、複写、削除を行なう。



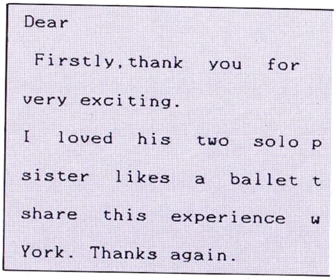
①作成画面

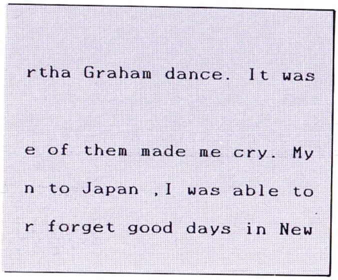


②sister の「si」まで入力



③ワードラップ機能





3 実用ソフトで多彩に活用

パソコンを便利な道具として使うなら、市販のソフトを利用するのが早道だ。MSX2には、バラエティに富んだソフトがそろっている。どれも家庭で役立つ本格派のソフトだ。パソコンは、多彩な用途に大いに力を発揮するにちがいない。

ソフトを紹介する前に、たいせつな注意をしておこう。

●フロッピーだから、こんな点に要注意

MSX2用のソフトには、カセットテープやROMカートリッジなどがあるが、ここでとりあげるのは、主として、3.5インチのフロッピーディスク。HX-34には、3.5インチのフロッピーディスク装置がついてるし、フロッピーなら内容の読みこみなどもスピーディに行なえるので、すぐに使える状態になる点もいい。

3.5インチのフロッピーは、プラスチックのケースに入っているので、あまり神経質になる必要はないが、シャッタを開けて、中のフロッピーにさわるのは厳禁。フロッピーはキズつきやすいし、キズがつくと、内容を正確に読みこむことができなくなる。

磁気を帯びたものを近づけないという注意も必要だ。フロッピーの内容が壊れるおそれがある。磁石だけでなく、スピーカーやテレビにも気をつけよう。

フロッピー装置へのセットは、電源を切った状態で、金属製のシャッタのほうから押しこむ。使い終わったら、フロッピーを抜いてから電源を切ること。使っている途中で引き抜いたりしないように。内容を壊さないための用心だ。

以下に紹介するソフトの一覧

商品名	概要	メディア(対象)	標準価格
漢字スプレッドシート Calc Mat	漢字表計算ソフト　横52列×縦128行 画面表示＝30桁　熟語辞書搭載	ディスク (MSX2)	14,800円
漢字データベース	カード式データベース　カードの設計・出力などを自由に設定、変更できる。(カードのサイズ31桁×8行　約500枚)	ディスク (MSX2)	19,800円
宛名書き（住所録）	漢字住所録　人名辞書、郵便番号による住所入力　ハガキ、宛名シール印刷	ディスク (MSX2)	17,800円
はがき君	年賀状、暑中見舞い等はがき印刷、定形句（謹賀新年等）、定形画（十二支）を読み出し、熱転写プリンタHX-P560で、はがきに印刷できる。定形画の修正及び漢字ROMを使えば定形句以外（住所等）も可能	ディスク (MSX MSX2)	9,800円
〈高校入試突破〉英単語完全マスター1200語	学研の教育ソフト　中学で習う1200語をマスターする。必修語、重要語、ハイレベル語より構成	ディスク (MSX2)	9,800円
すいすいローラー（さんすう）	学研の教育ソフト　四則演算の基礎を修得	ROM (MSX MSX2)	4,800円

●気になる画面はすぐにプリント

ソフトを使っているとき、画面の表示を紙に印刷しておきたいことがある。たとえば、ゲームで高得点を出したとき。スコアが表示されている画面をそのまま印刷しておけば、上達の記録にもなるし、友人に自慢もできる。それが、簡単なキー操作でできるんだ。

■画面プリントは CTRL + COPY キーで

キーボードを見てほしい。長い空白キーの右横に赤い字で「COPY」と書かれたキーがあるはずだ。これが画面をそのまま印刷するためのキーだ。印刷しておきたい画面になったら、接続してあるプリンタ（HX-P550かHX-P560）の電源を入れて、CTRL を押しながら COPY キーを押す。これで、画面の表示内容が紙に印刷される。

このように、画面を印刷したものをハードコピーという。

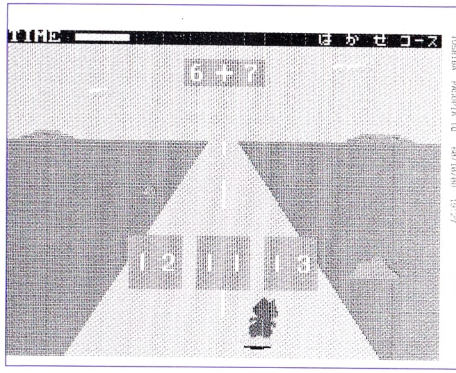
■キーひとつでこれだけ使える

ゲーム以外にも、使い方はいろいろある。思いつくままにあげてみよう。

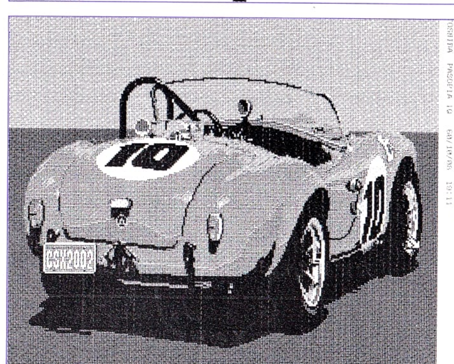
学習ソフトで勉強しているなら、むずかしかった問題や理解しにくかった部分などをプリント。あとでじっくり復習しよう。パソコン学習の効果は、いっそう確かなものになるだろう。

自分でつくったグラフィックスなども、ハードコピーしておくといい。写真という手もあるが、こちらのほうが簡単だ。いくつも集まれば、グラフィックスのオリジナル作品集ができあがる。

プログラム自作派なら、プログラムの内容をチェックするのに使えるぞ。プログラムを実行して、その結果をハードコピーする。それを見ながら、細かいところまで調べて手なおしする。画面を見ているだけでは気がつかないところが、きっといくつかあるはずだ。このくり返しで、質の高いプログラムができあがるだろう。







●読みやすい漢字で表計算もラクラク

漢字スプレッドシート

たくさんの数値を表の形にまとめて、縦・横を合計したり、平均を計算したり。ときには、数値の小さい順や大きい順に並べかえたり——手作業ではたいへんな仕事をスイスイこなしてくれるのが、「漢字スプレッドシート」だ。

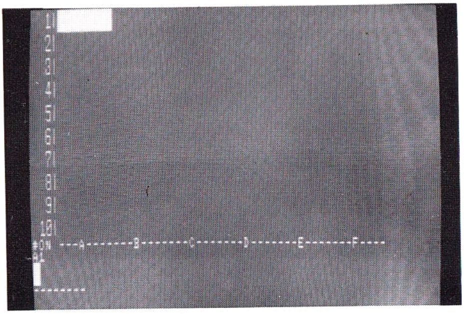
各種の命令や関数（SUM→合計、AVR→平均など）が利用できるので、複雑な表やデータづくりもオーケーだ。

■計算・並べかえ・修正はすべて自動

このソフトを使うと、計算式さえ教えてやれば、めんどうな計算や並べかえはすべて自動的に行なわれる。まちがったところを修正すると、その数字に関係のあるところはすべて修正される。手作業だと、あちこち計算しなおすのでひどく手間がかかるが、これなら一瞬にして完了だ。

漢字を使えるから、わかりやすい表ができるし、熟語が2万語入っているから、単漢字変換と合わせて漢字の入力も簡単。

応用範囲はいろいろだ。キミなら統計や実験結果を表にしたり、成績表をつくったりするのに使える。ママなら家計簿、パパなら仕事で使う表ができる。



❶最初の表

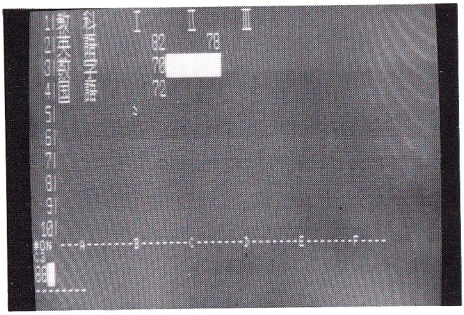
■成績表をつくる

さっそく使ってみよう。キミの成績表をつくることにする。「漢字スプレッドシート」の表は、横52×縦128行から構成されているので、項目の数は自由に設定できる（❶）。

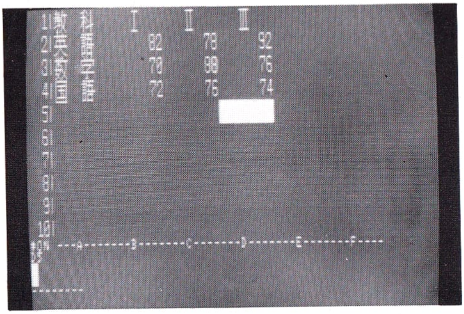
この表に項目名と数値を入力すれば、表ができあがる。項目は縦に教科名、横に試験の順を並べることにしよう（❷）。

項目名の入力は、ワープロと似たような操作でいい。各マス目に必要事項を書きこんでいき、一つのマス目を書き終えるごとに RETURN を押す、という作業をくり返すと、表ができあがる（❸）。

これがスプレッドシートだ。



❷項目と数値を入力



❸表の完成

●パソコンでユニークな年賀状

はがき君

年末がくるたびに年賀状に悩んでいる人には、うれしいソフトだ。文字はもちろん、絵も自由に描いて、ハガキに印刷できる。絵が苦手なら、登録されているレディメイドの図案を使えばいい。もう、今年から、ユーウツな年末とはおさらばだ。

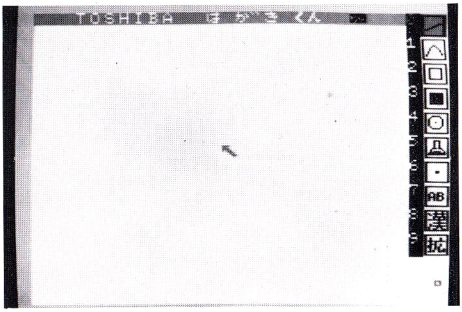
■文案も図案も登録ずみ

最初の画面(❶)は、画用紙のようなもので、ここに自由に絵を描ける。が、まず、このソフトに登録されている文案や図案を使うほうがてっとり早い。

画面の右に縦に並んでいるのが機能のメニュー。目的の機能に赤いカーソルを合わせて空白キーを押せば、その機能が選択される。

さて、登録された図案や文案を使うには「9拡」を選択して、さらに別の機能メニューが出たら「4L」(LはLOADのこと) を選択。あとは、ひらがなで、必要な文案なり、図案なりの名前を入力して呼び出す。

文案は「謹賀新年」や「賀正」など、年賀状の決まり文句がある。図案は、十二支のそれぞれの絵が入っている。たとえば虎の絵は、「とら」と入力して RETURN を押すと表示される(❷)。



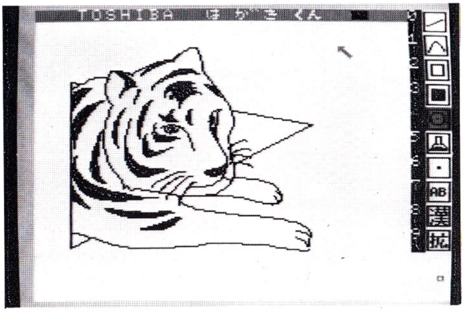
❶初期画面

■呼び出した文案・図案を好きな位置に

このようにして、文案・図案を呼び出したら、今度は、適当な位置に移動する。文字と図の配置のぐあいをよく確かめよう。なおしたあとは残らないから、何度でも変更できる。

配置が決まったら、あとは「5P」(PはPRINTのこと) を選んで印刷する。これが、もっとも手軽な年賀状だ。

少しくふうするなら、これに絵の具などで色をつける。それだけで、見ちがえるようになるだろう。



❷「とら」の図案を表示



■平均点を計算する

前ページでつくった表をもとに、各教科の平均点を計算してみよう。

これを自動的に計算してくれるのが、AVR関数だ。最後の国語は、B4～D4までの平均だから、「+AVR(B4，D4)」と入力すればよい(❶)。これで RETURN を押せば、画面の大きいカーソル（ブロックカーソル）の位置に、平均値が表示される。

■一部を変えると全データが修正される

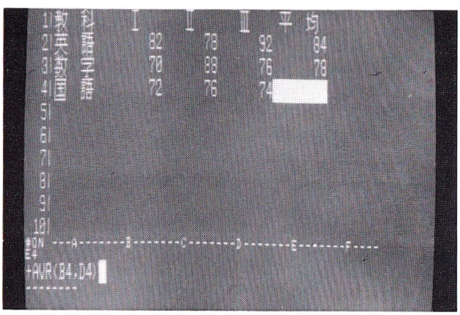
ここで、第1回の試験の英語の点数を変えてみよう。英語の82点を85点にして、RETURN を押す。これで、英語の平均点が84点から85点に自動的に変わった(❷)。第1回の平均点や合計、全体の平均点や合計など、変更した数値に関係のあるところがあれば、それらもすべて修正される。

この機能は、誤りを修正するのに便利なばかりでなく、「もしあのとき何点とれたら」とか「こんどは、これを何点とれば全体の平均点があがる」などと模擬的に結果を計算してみることもできる。

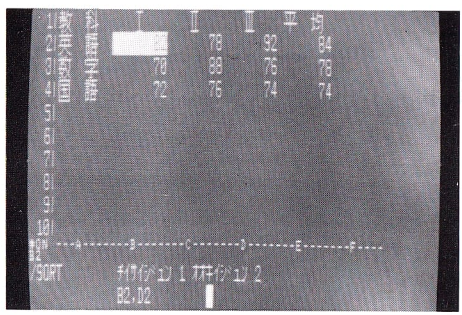
■平均点の高い順に並べる

試験の点数を、大きい順や小さい順に並べかえてみよう。それには「／SORT」命令を使う。それを打ちこんで範囲を指定すると、大きい順か小さい順かを聞いてくる（❸ここでは英語の数値を入れかえる)。大きい順なら、2 RETURN で並べかえが行なわれる(❹)。

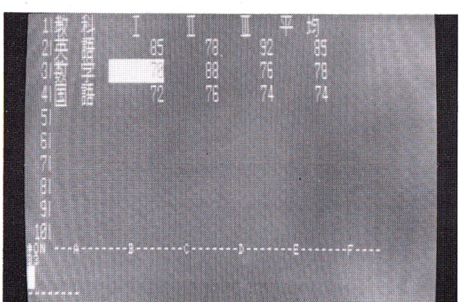
漢字スプレッドシートには、このように表づくりや計算に必要な命令や関数がそろっている。データの追加、修正、削除なども朝メシ前。つくった表や計算の結果を印刷したり、フロッピーに保存しておくことも可能だ。

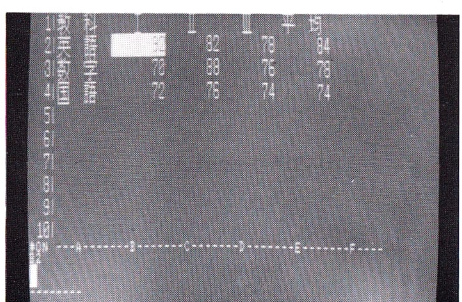


❶平均点の計算



❸数値の並べかえ


❷数値の変更


❹並べかえ完了

■オリジナルな絵でキミだけの年賀状

人がつくったものより、自分だけしかできない年賀状を出したい——そういうキミには、ぜひ、このソフトのグラフィックエディター機能を使いこんでほしい。

カーソル移動キーで青いカーソルを移動して、好きな絵が描ける。直線でも四角でも円でも、２点の指定だけで即座に線が引かれる。

複数の点を結ぶ自由な曲線を描くこともできるから、思いどおりの絵が、すばやくできあがる(❶)。

画面右下には、カーソルのある部分が拡大されているので、点ごとに修正ができる。細かいところにもキミのセンスが生かせるぞ。

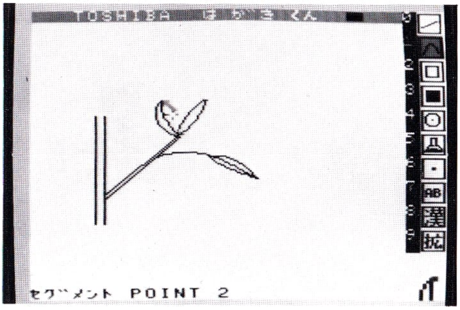
漢字も普通の大きさと倍の大きさのどちらかを選ぶ。「謹賀新年」は大きな字、住所は普通の文字というように書き分けられる。もちろんアルファベットや数字も入力できる（❷～❹)。

まったくのオリジナルがたいへんなら、登録してある図案を呼び出して手を加えればいい。ちょっとしたくふうで、もとの図案とは感じの違うものができあがる。

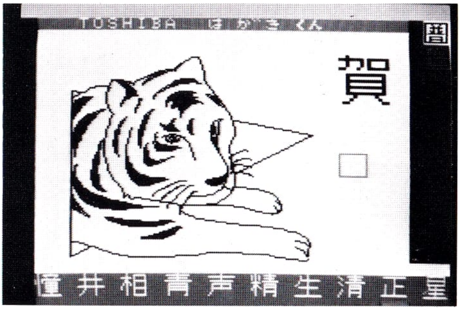
■気に入ったものはフロッピーに保存

いろいろな図案をつくって、気に入ったものはフロッピーに保存しておくといい。画面右の機能メニューで、「９拡」を選んで、次のメニューで「3S」(SはSAVE(セーブ)のこと)を選べばいい。

これで、画面の下には「SAVE NAME」と出るから、登録したい図案の名前を決め、ひらがなかカタカナで入力して RETURN。すぐに登録は終了する。呼び出しは、同じメニューの中の「4L」でできる。



❶直線や曲線を描く



❸賀正を入力



❷謹賀新年を入力



❹住所や電話番号を入力

●学習ソフトならゲーム感覚で実力アップ

すいすいローラー(さんすう)&高校入試突破

パソコンを使った勉強でいいのは、マイペースでできることだ。わからないところは、何度でもくり返して学習できるし、得意なところは短時間ですませばいい。気分よく勉強できて、確実に力をつけることができる。そんな学習ソフトを二つ紹介しよう。

■すいすいローラー(さんすう)

計算問題に正確に速く答えるためには、くり返し学習することがいちばんだ。このソフトなら、ゲームのような気分で楽しみながら練習できるから、幼稚園児から、計算が不得意な小学校高学年まで使え、計算力はみるみる向上するはずだ。

まず問題をメニュー画面から選ぶ。学校の進度に合わせた問題を選んだり、不得意なところを重点的に練習するといい。

答えるスピードは、入門、秀才、博士、天才の四つのコースから選ぶ。

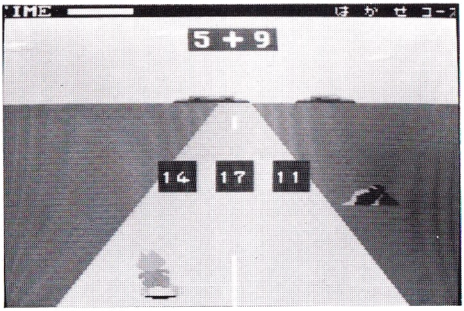
コースごとに時間制限があるから、問題が出たら、すばやく答えることが必要だ。ローラースケートをはいたネコが正しい解答の上を走るようにコントロールする(❶)。コースからはずれたり(❷)不正確のボードにぶつかると、減点されたうえに時間をロスして、不利になるぞ。

さて、ガンバッて、時間内に全部の問題に答えよう。得点が80点以上だと合格だ。

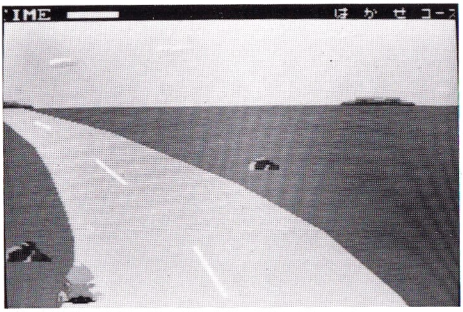
制限時間を超えると、不合格でゲームオーバー。また、全問答えても80点未満だと、やっぱり不合格だが、この場合は、リターンマッチができる。リターンマッチでは、まちがえた問題や時間をロスした問題だけ出題されるから、学習効果もバツグンだ。これで不合格だと、ゲームオーバーになる。

軽快なバックグラウンドミュージックに乗って、名前のとおりス～イス～イとお勉強。

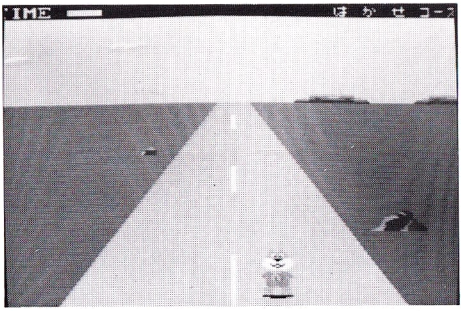
ネコちゃんは、合格したときに振り向いてくれる(❸)。笑顔がカワイイぞ！



❶ゲームで学習



❷ネコが転んだ！



❸ゴールイン(合格)

英単語完全マスター1200語

■高校入試突破　英単語完全マスター1200語

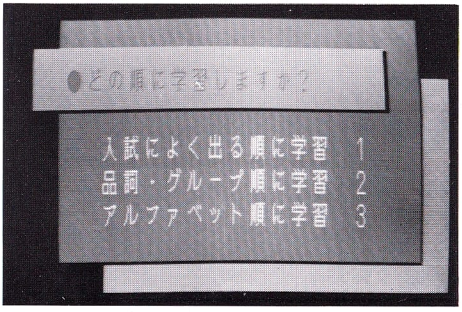
高校入試突破のために、中学の３年間で学習する英単語1200語を、完全にマスターするための学習ソフト。英単語のマスターもくり返しが重要だが、単純なくり返しでは退屈だし、力もつきにくい。このソフトは、単調さをなくし、実力の養成や点検が確実に行なえるように、いろいろくふうされている。

まず、全体を必修語(400語)、重要語(400語)、ハイレベル語(400語）と、むずかしさによって3グループに分けている。

さらにそれぞれのレベルで、「入試によく出る順」「品詞・グループ順」「アルファベット順」の３コースから選ぶ。自分のレベルや好みに合ったコースを選べるわけだ(❶)。

グループとコースを選んだら、いよいよ学習開始だ。その日にマスターする単語が表示され、各語ごとにスペル、意味、発音、アクセント、注意点、関連事項、反意語、対語などを確認する(❷)。

次に「どうしますか?」の選択で(❸)、よければ、意味ドリル(選択肢)→つづりドリル(穴うめ完成)と進む。各ドリルが終わると、得点と合否、まちがえた語が表示される。まちがえた語はくり返し練習して、完全にマスターしたら、しあげのテストに挑戦することになる。

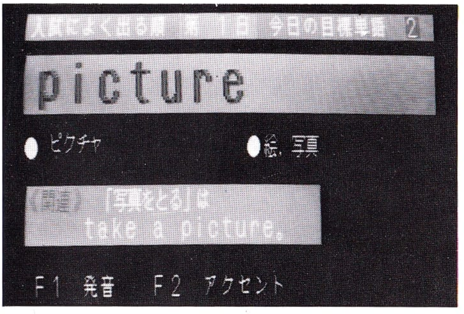


❶コースの選択

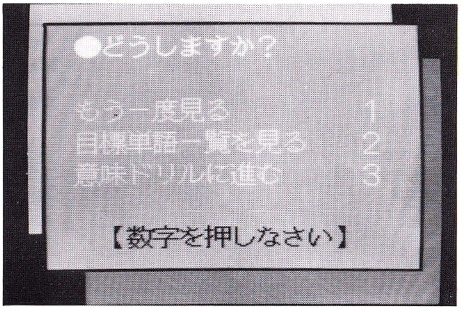
このテストでは、わからない問題はあとまわしにできるし、気づいたまちがいは修正できる。しかも、全問答え終わってから採点キーを押して、一括採点するから、まるで、実際のテストのようだ。

さらに６日ごとに、まちがった単語を中心としたテストがある。

こうして、記憶のあいまいな単語は、どんどん減っていく。夢中でソフトにとり組んで、ひととおり終わるころには、完全にマスターできているはずだ。



❷各単語の学習



❸進路の選択

●たくさんのデータを電子カードでためこもう

漢字データベース

情報の整理は、カードを使うと、便利な点が多い。追加も削除も簡単だし、並べかえや分類も自由にできる。必要なカードだけとり出すなんてことも可能だ。

これをパソコンでやってしまおうというのが、この「漢字データベース」だ。

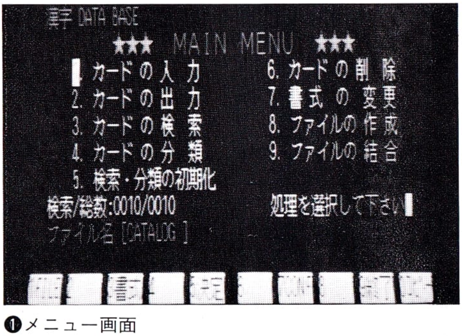
■カードを設定して、データを入れる

住所録や献立カード、レコードや本の管理など家庭での利用法のほかに、顧客管理、生徒の管理、名刺の管理など、このソフトも利用範囲が広い。

まずはメニュー画面(❶)から、「1.カードの入力」を選ぶ。1枚のカードにどんな項目を入れるか、それをどう組み立てるかは、使い方によって違う。住所録なら名前や住所、郵便番号、電話番号などだが、献立カードなら、料理名、材料、作り方といった項目になるだろう。このような違いに対応するため、カードは自由に設定できるようになっている。この設定しだいで、使いやすさが違ってくるぞ。

1枚のカードは1画面で、横31字×縦 8行の大きさ以内で設計する。ここでは、レコードライブラリで考えてみた(❷)。

■内容はワープロ感覚で入力



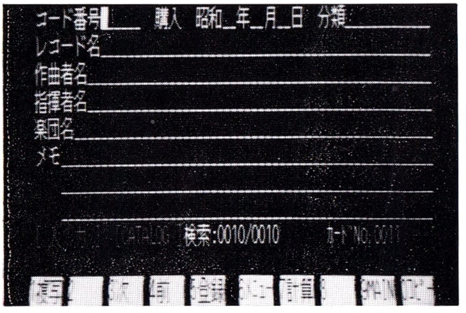
❶メニュー画面

カードができたら、1枚ごとに内容を入力することになる(❸)。このソフトは、英数字、カタカタ、ひらがな、特殊文字などのほかに漢字も使えるから、読みやすい。

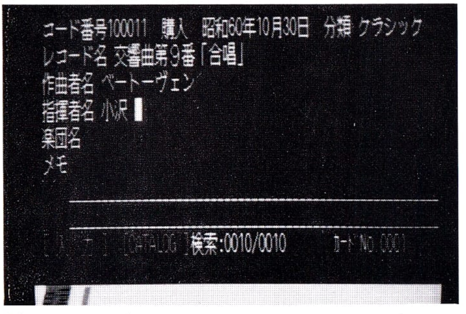
漢字は、単漢字変換だが、熟語ROM(ロム)カートリッジを使えば、文節変換で入力はラクだ。1枚のフロッピーには約500枚のカードが保存できる。レコードライブラリでも住所録でも、家庭で使うのなら、これだけの量が保存できれば十分だろう。

500を超えるデータを扱うなら、フロッピーを増やせばいい。

もちろん、変更や追加、削除なども自由自在だ。



❷項目の設定



❸データを入力

■保存したデータは自由に引き出せる

フロッピーに保存したデータは、いろいろな形に引き出して使える。条件を与えると、自動的に条件に合うカードが抜き出されて、画面に表示される。並べかえも自由だ。しかも速い。

たとえば、ある楽団の演奏したものだけをピックアップしたり、購入した順に並べたり、紙のカードやノートでは、かなり手間のかかることも軽々とやってのける。

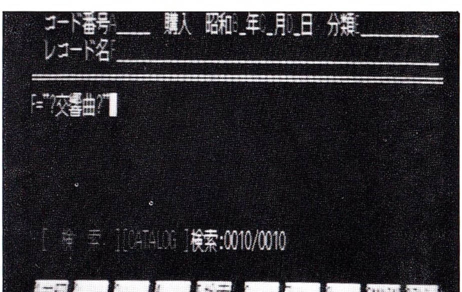
条件によるカードの抜き出しなら、メニュー画面で「3.カードの検索」を選んで、検索の画面になったら、希望する条件を入力すればいい(❶)。

これで、条件に合った人のカードはすぐに画面に表示される。条件に合うカードが何枚もあれば、↓キーで次々に出てくる。

■並べかえで意外なことがわかることも

メニューで「4.カードの分類」を選ぶと、並べかえができる。分類の画面で、分類項目を設定すればいい。たとえば購入年月日の順に並べかえるとしよう。分類項目で「BCD」を入力して(❷) RETURN 。次に並べかえの順序を聞いてくるから、どちらかを選ぶ。

単に保存したデータを引き出すだけでも、けっこう便利に使えるが、それ以外に、次のような利用法もある。ある条件で抜き出したカードを、別の基準で並べかえる。たとえば一定の時期に買ったレコードを、作曲者別に分類したらどうだろう。いままで気がつかなかったことがわかるかもしれない。



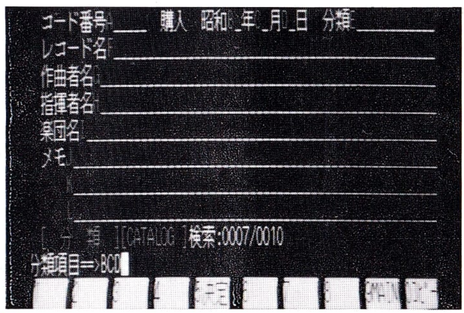
❶「交響曲」だけを検索

こんなこともできるのが、データベース・ソフトのおもしろさだ。

■ある項目だけを抜き出して一覧表もできる

カードはもちろん、プリンタで印刷できる。カードをそのまま印刷するのではなく、必要な項目だけを抜き出して一覧表にする、という方法もある。

たとえば購入時期別に、引き出したカードからレコード名と作曲者名を抜き出して印刷、なんて芸当もオーケーだ。



❷購入年月日の順に並べかえ

番号	F
0001	交響曲第９番「合唱」
0002	交響曲第５番「運命」
0003	交響曲第４０番
0004	幻想交響曲
0005	交響詩「海」
0006	交響曲第９番「新世界」
0007	ピータと狼
0008	組曲「惑星」
0009	交響曲 第９番「合唱」
0010	交響曲第７番

●一度つくれば宛名書きの自動化が可能

宛名書き

年賀状や暑中見舞い。書く気持ちはあるけれど、宛名書きがめんどうで、つい出しそびれる——なんて人は多いはず。宛名書きを自動化して、楽になったぶん、内容に気持ちをこめる。こうなれば、理想的だ。「宛名書き」を使うと、それができる。

■宛先の住所は画面上で入力

日付を入力して、プリンタの機種を選択すると、メニューが表示される(❶)。ここで「1.［データ入力］」を選べば、宛名などの内容を入力できる状態になる。ひとり1画面のカード形式で、郵便番号、電話番号、住所、氏名、生年月日、星座などを入力して、記録しておける。

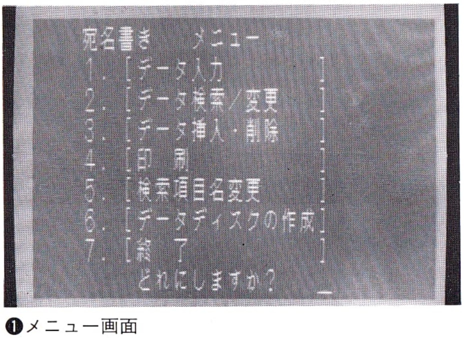
操作法はワープロと同じだが、人名の姓が約1500、名が約1500、それに郵便番号に対応する住所が約3400、熟語として登録されている。これを使うと、人名は熟語変換で入力できるし(❷)、住所は郵便番号を入力すると自動的に入力される。生年月日を入れると、星座も自動的に出る。とても便利だ。こうして、どんどん入力していこう(❸)。

宛名はハガキにも印刷できる。宛名シールに印刷すれば、ハガキでも封筒でも、裏の紙をはがしてはりつけるだけですむ。

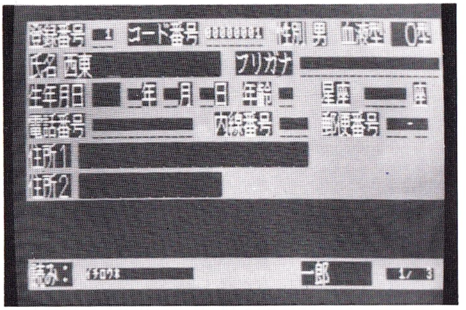
■特定の人の宛名だけを印刷できる

いつも、住所録にある人全員に手紙を出すとは限らない。住所録の中から、ある条件に合う人に出すこともある。そんなときには、条件に合うカードを抜き出してから、印刷すればいい。たとえば北海道に住む人にだけ手紙を出す、血液型や星座で分類して印刷する、なんてことも簡単だ。

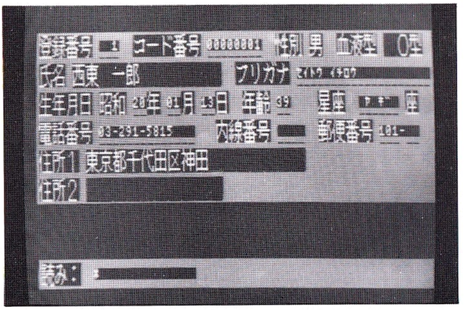
1枚のフロッピーには、約500人の宛名を保存しておける。キミが毎年出す年賀状の数の何倍かになるだろう。これからは、きっと、登録する人の数がだんだん増えていくのが、楽しみになるぞ。



❶メニュー画面



❷熟語読みで人名を入力



❸入力中の画面

4 夢が広がるパソコン通信

●RS-232Cと音響カプラでテレコンピューティング

パソコンは、ひとりで使うのもいいけど、いろいろな人とコミュニケーションがとれたら、もっとおもしろくなる。パソコンが普及して、いま、それが可能になってきた。遠く離れたパソコンどうしをつないで、お互いにやりとりをすれば、パソコンの世界はグンと広がる。これがテレコンピューティングだ。

パソコンどうしをつなぐのには、主に電話回線が使われる。RS-232C インターフェースを用意し、音響カプラをパソコンにつないで、そこに電話の受話器を置く。もちろん、相手のパソコンも同じようにする。これでオーケー。

パソコンから出た電気信号は、音響カプラから電話回線に入り、相手に届いてパソコンで再生される。ひとことでいえば、パソコンどうしが電話をかけているようなものだと思えばいい。

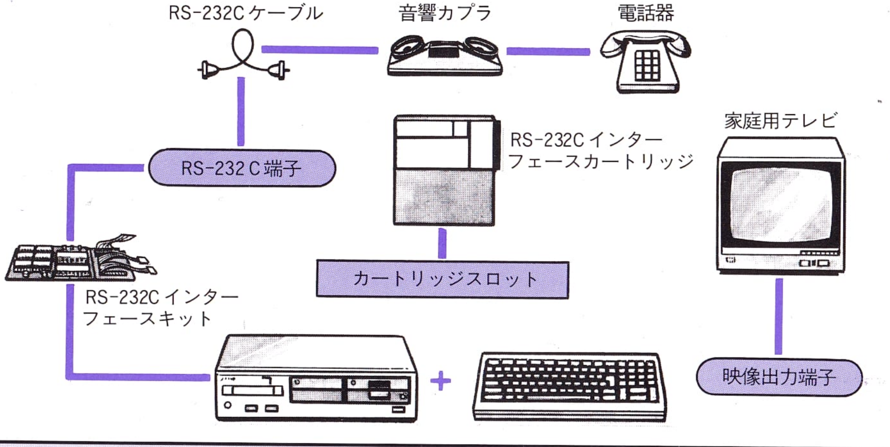
電話を使っているように、どんなに遠くでもすぐにつながるし、お互いに話をやりとりできる。外国のパソコンとやりとりすることだって可能だ。

この機能を使うと、いままで思いもしなかった世界が開けてくる。いろいろな情報サービスをする組織の会員になると、キミの机の上のパソコンは、日本全国、いや全世界と直結するんだ。

テレコンピューティングを使うと、ニュースは、新聞やテレビよりも速く届く。じっさいに事件が起こった現場にいる記者に質問をして答えてもらう、なんていうことも可能だ。知りたい情報が即時に手に入る。

旅行や観劇、ショッピングなどの情報なら、暮らしにすぐに役に立つ。飛行機やホテルの予約をしたり、パソコンで買い物をして、支払いもすます。家にいながら、すべてができるようになる。

会員に呼びかけて、いろいろ貴重な情報やアドバイスを提供してもらったり、外国にパソコン・フレンドをつくったり、と楽しみはきりがないぞ。



プログラムやデータの交信をする

身近な例でも、テレコンピューティングを大いに利用したい。日本語ワープロ「漢字君」でつくった文章を電話で送ることは、前に話した。パソコンで送って、プリンタで印刷してもらえば、速く、確実に送れる。手紙や電話に次ぐ、通信手段だ。電話で伝えるには内容がこみいっている、手紙だと時間がかかりすぎる、という場合にはうってつけだ。ワープロでつくった表や図も、スピーディに相手に届く。

恋愛中のキミなら、パソコンによるラブレター大作戦などは、いかが。相手へのアピール度も高いはずだ。ピッ、ピッ、ピッというキー操作で交流が深まるなんて、テクノ時代にふさわしいつきあい方だと思わないか。

いろんな面で、コミュニケーションのあり方が少し変わってくるかもしれない。

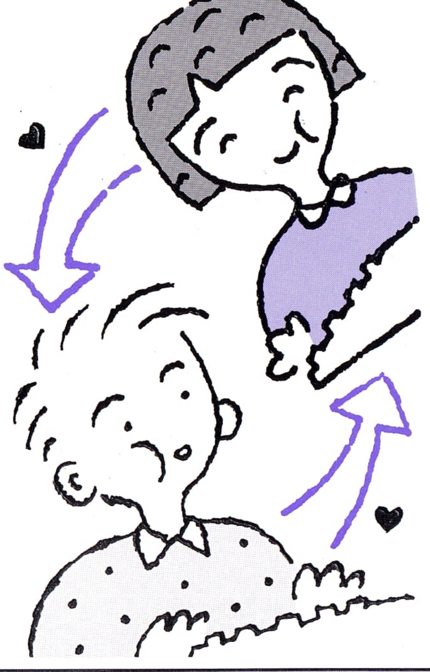
この通信機能、ワープロ文以外にも、いろいろに使える。たとえば、プログラム。できたばかりのプログラムを友人に送る。実行してもらって感想を聞く。もの足りないところは、アドバイスしてもらう。

プログラムをつくっている途中で、どうしてもアイデアが浮かばなくなったときには、友人に知恵を借りる。作成中のプログラムを送って、手を入れてもらい、コメントをつけて返送してもらおう。

自慢の自作プログラムをやりとりしたり、友人のつくったゲーム・プログラムを送ってもらったり。本や雑誌に出ているプログラムを、グループで手分けして入力、電話で送るようにすれば、キミのゲーム・プログラムはどんどん増えていくだろう。

数人で別々につくったプログラムをドッキングする、なんてこともできる。やり方によっては、思いがけないすごいプログラムになるかも。予想どおりにしあがるかどうか、ちょっと知的なゲームとして楽しむのもおもしろい。

データをやりとりしたり、いろんなデータを集めて、みんなで使えるようにしたりすることもできる。クラスのアイドル情報局、期末試験ヤマカン情報、遊び場データベースなどなど。アイデアしだいで、いろいろ可能だ。



●電子掲示板で広げたい パソコン・フレンド

電子掲示板というシステムを知っているかな。BBS（Bulletin Board Systemの略）ともいい、キミのパソコンで書きこんだり、読んだりできる、一種の掲示板だ。この掲示板を主催する組織の会員になれば、だれでも自由に掲示板を使うことができる。メッセージを送ったり、掲示板の内容を読んだり、掲示板で交わされる議論に加わったり……ずいぶんいろいろな楽しみ方ができそうだ。

このBBS、アメリカではとても盛んで、数多くのシステムがあるそうだ。日本では、まだそこまではいかないが、それでも、各地にぼちぼち広がりつつある。今後は、もっとハイテンポで増えていくにちがいない。

電子掲示板には主宰者がいる。企業や趣味のグループ、個人など多彩で、目的もさまざま。会員になるにも有料のところと無料のところがある。こんなことは、事前に調べておこう。

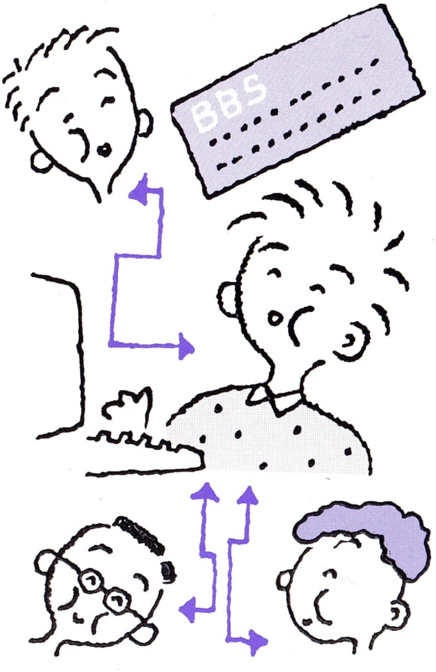
電子掲示板を利用するには、パソコンのほかにRS-232Cキットと音響カプラが必要だ。あとは、利用したいシステムの主宰者に申しこめばいい。手続きをすませば、BBSフレンドに仲間入りだ。

掲示板には、いろいろな情報がのっているはずだ。それらを読んで、役に立ちそうな情報を手に入れよう。

反対に、たとえば、「○○について教えて」「○○譲ります」など、自分のメッセージを書きこむ。悩みごとの相談にのってくれる人を探す、なんてこともできる。旅行に出かけるときなど、目的地にくわしい人に呼びかけて、アドバイスを受けるのもいい。

机の上のパソコンは、もはや小さな世界にとどまってはいない。キミは、パソコンを通じて、多くの人に接することができる。だれかな、「ガールフレンド募集中」などと書きこんでおこうなんて考えているのは。

そうそう、アメリカのデータベース・サービスの会員になるなんてことだってできてしまう。アメリカで起きたできごとを新聞より早く知ったり、アメリカの最新パソコン情報を調べたり。もちろん英文だから、ついでに英語の勉強もしちゃおう、というのはよくばりかな。



PASOPIA IQ 仕様一覧

項目		仕様			
形名		HX-30	HX-32	HX-33	HX-34
標準価格		￥43,800	￥79,800	￥99,800	￥148,000
CPU（クロック）		Z 80 A（約3.58MHz）			
メモリ	ROM	32KB（MSX-BASIC）	32KB（MSX-BASIC）＋32KB（内蔵ソフト）	48KB（MSX-BASIC Ver. 2.0）＋80KB（内蔵ソフト）＋128KB（漢字）	48KB（MSX-BASIC Ver. 2.0）＋80KB（内蔵ソフト）＋128KB（漢字）＋16KB（MSX DISK BASIC）
メモリ	メインRAM	16KB	64KB		
メモリ	ビデオRAM	16KB		128KB	
プログラム言語		MSX-BASIC	MSX-BASIC、通信用MSX-BASIC、拡張BASIC（プリンタスプーラ機能、RAMディスク機能）	MSX-BASIC Ver.2.0、通信用MSX-BASIC、拡張BASIC（プリンタスプーラ機能）	
画面表示	VDP	T 6950		V 9938	
画面表示	テキスト	32桁（初期29桁）×24桁／40桁（初期39桁）×24行		最大80字×24行／40字×24行	
画面表示	グラフィック	横256×縦192ドット		最大横512×縦212ドット	
画面表示	スプライト	32画面			
画面表示	カラー	16色		最大256色	
高解像度グラフィックモード		256×192ドット、16色		256×192ドット、16色（512色中）	
低解像度グラフィックモード		64×48ブロック、16色		64×48ブロック、16色（512色中）	
ビットマップグラフィックモード		—		512×212ドット、16色（512色中） 256×212ドット、256色	
サウンド	PSG	AY-3-8910相当			
サウンド	機能	8オクターブ3重和音＋1効果音			
キーボード	キープラス本体	一体			セパレート
キーボード	文字記号種類 キー配列	73キー、英数字、英記号、ひらがな、カタカナ、かな記号、グラフィック記号		74キー、英数字、英記号、ひらがな、カタカナ、かな記号、グラフィック記号	同左＋テンキー、四則演算キー
キーボード	キー配列	英数かな共 JIS配列準拠			
内蔵インタフェース	オーディオカセット	FSK方式 1200／2400ボー			
内蔵インタフェース	プリンタ	セントロニクス仕様準拠			
内蔵インタフェース	通信機能	（別売カートリッジHX-R700）	HX-R701内蔵可（別売）	HX-R702内蔵可（別売）	HX-R703内蔵可（別売）
時計・カレンダー機能		—		あり	
漢字ROM		—	内蔵（JIS第一水準）		
3.5インチFDD		—（別売）			1台内蔵、2台内蔵可
画像プリント出力		—		あり	
外部端子	カートリッジスロット	2個（上面・背面）			2個（前面・背面）
外部端子	RF端子	1個			
外部端子	映像端子（コンポジット）	1個			
外部端子	アナログRGB端子	—		1個	
外部端子	音声端子	1個（モノラル）	2個（ステレオ）		
外部端子	レコーダ端子	1個			
外部端子	ポート端子（ジョイスティック）	2個			
外部端子	プリンタ端子	1個			
外部端子	RS-232C端子	—			
外形寸法		幅420×奥行220×高さ75mm			本体 370×300×90mm キーボード 370×170×45mm
重量		約2.8kg			本体 約5kg キーボード 約1kg

入門機から本格タイプまで
PASOPIA IQ バリエーション

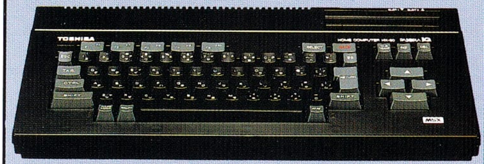




■HX-33　64Kバイト MSX2
日本語ワープロソフト、漢字ROM内蔵



■HX-32　64Kバイト MSX
日本語ワープロソフト、漢字ROM内蔵



■HX-30　16Kバイト MSX
ゲームから学習まで、手軽な入門機



■HX-22　64Kバイト MSX
日本語ワープロ、通信機能を内蔵



■HX-21　64Kバイト MSX
ワープロ内蔵。映像３出力で鮮明画像



■HX-20　64Kバイト MSX
ワープロ内蔵のスタンダード・タイプ

●本文で紹介したゲーム

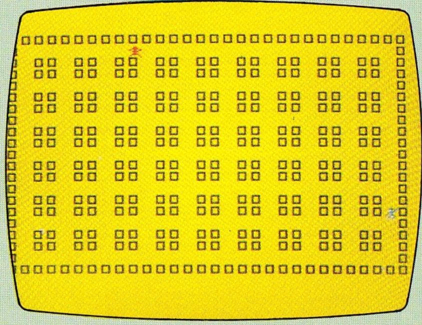
逃げろや逃げろ

本文　102ページ

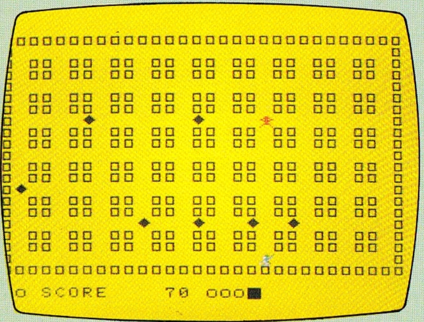
敵は赤いインベーダーで、自分は青い人間。スタートとともに敵が追いかけはじめるから、逃げろや逃げろ。つかまりそうになったら、◆のバリケードをつくる。敵がそれを食べているあいだに、どんどん逃げよう。



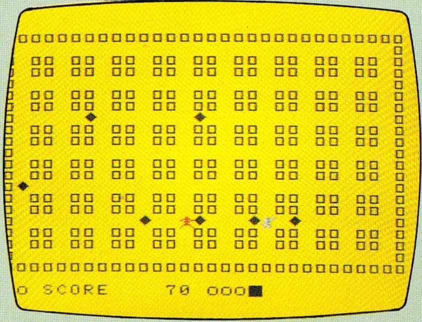
❶最初の画面。敵のパワーを聞いてくる。とりあえず、10としてみよう。



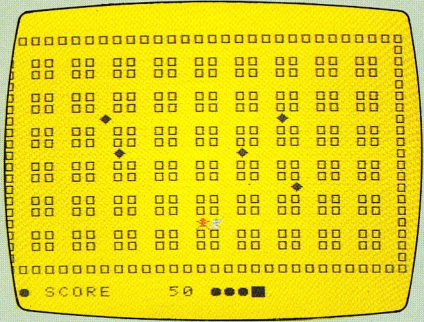
❷ゲーム開始。インベーダーは最短距離を追いかけてくるから、うまく逃げろ。



❸バリケードをつくった状態。つくりすぎると、逃げ道も少なくなるぞ。



❹敵がバリケードを食べているところ。近づいてきた、早く逃げよう。



❺あーあ、ついに追いつかれた。残りのバリケードの数が得点（SCORE）になる。

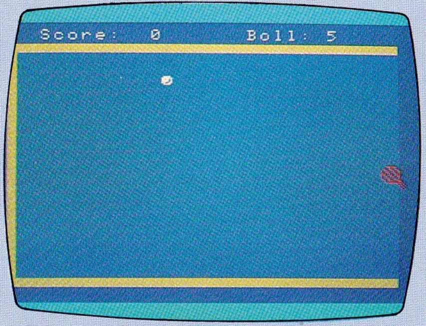


❻ゲーム終了の画面。得点が表示される。さあ、もう一度挑戦だ。

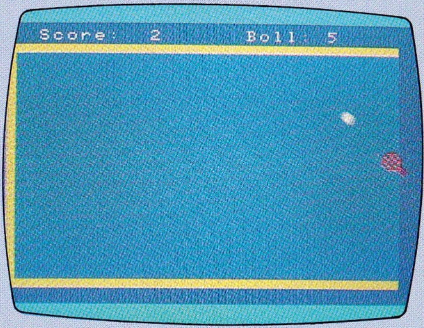
スカッシュゲーム

本文　135ページ

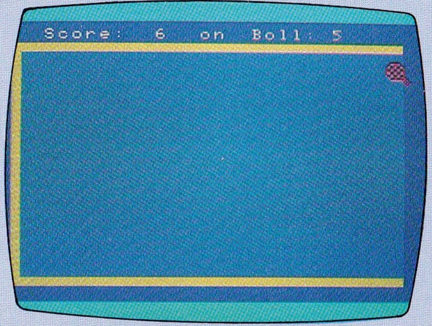
壁に囲まれたコートでボールを打つ、テニスのようなゲーム。得点が増えるにつれて、ボールが見えなくなったり、障害物があらわれたりと、だんだんむずかしくなる。ボールの数は5個。なくなったらゲーム終了だ。



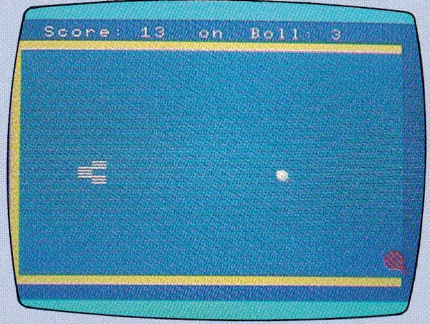
❶最初の画面。ボールが動きはじめたところ。ラケットを上下に動かして、ボールを待とう。



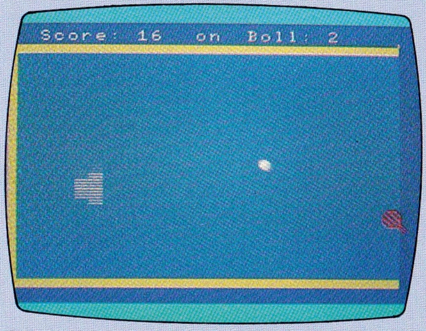
❷うまい！　ボールをみごとに打ち返したところ。でも、このへんはまだ簡単だ。



❸得点が6点以上になると、画面中央にきたボールが見えなくなる。どこから出るかな？



❹11点以上で、障害物があらわれる。これに当たると、ボールの動きが不規則になるぞ。



❺オッ！　障害物が動きだした。ボールがどうはね返るか、それを読むのがポイントだ。

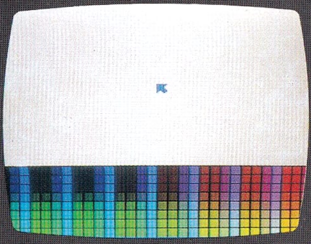


❻ボールを使いきって、ゲーム終了。21点以上になると、さらにむずかしくなる。

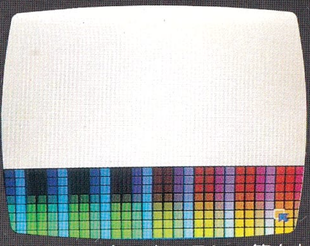
線画遊び

本文　163ページ

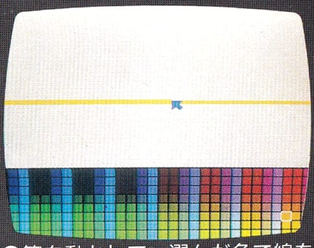
256色が使える、お絵かき遊び。色鉛筆でも使うように、簡単に絵が描ける。コンピュータ・グラフィックスに挑戦だ。



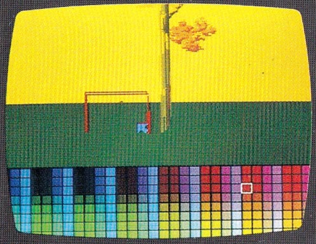
❶最初の画面。↖が絵筆で、下が絵の具だ。



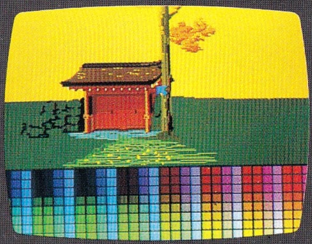
❷使いたい色のところに、筆をもっていく。



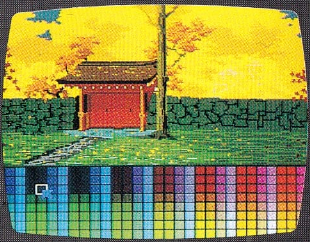
❸筆を動かして、選んだ色で線を引く。



❹かなり進んだところ。何ができるかな?

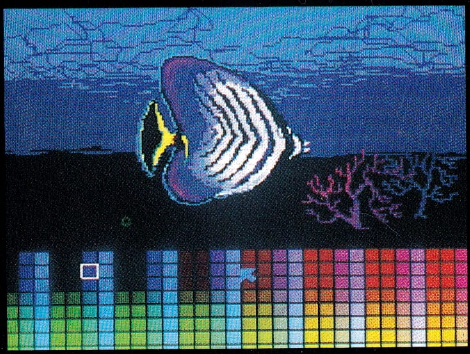


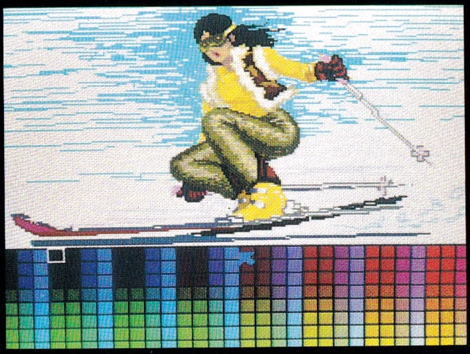
❺りっぱな門と、木ができた。



❻背景を細かく描いて完成。

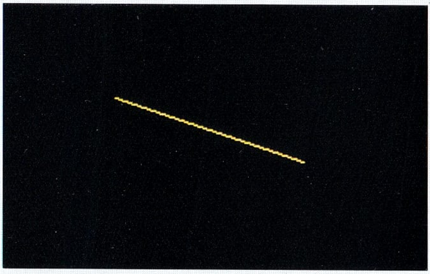
⊙同じようにしてつくった作品例



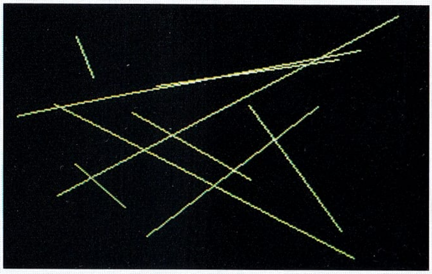


PART⑧ パソコン・グラフィックスの実行結果

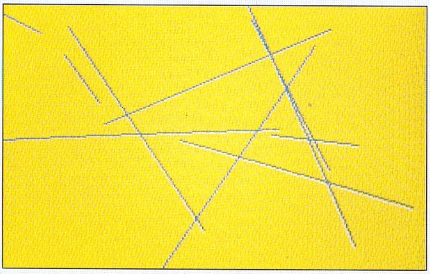
直線を描く
（170ページ）



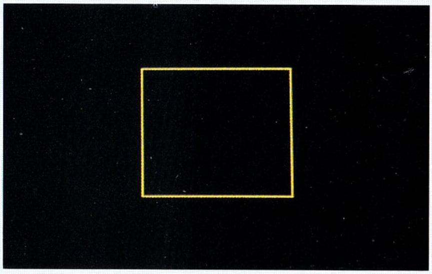
10本ごとに色を変えながら線を描く（171ページ）



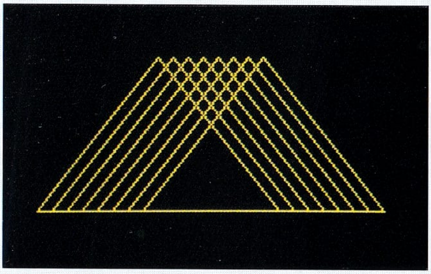
背景の色、線の色を決める〔10と4〕（172ページ）



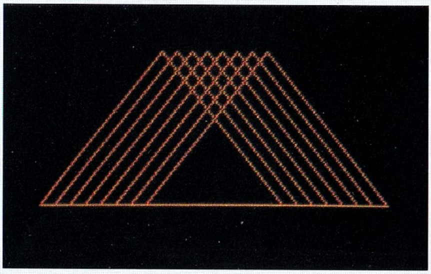
黒地に黄色の四角形を描く
（173ページ）



三角形が横にずれて、色が変わる
（175ページ）



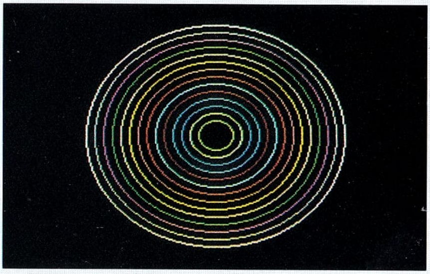
↓



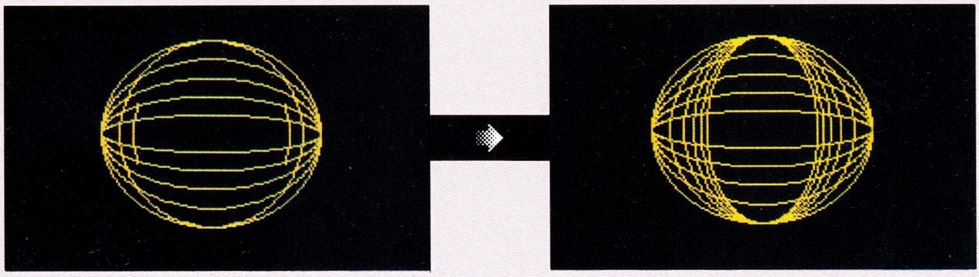
同心円を次々に別の色で描く
（176ページ）



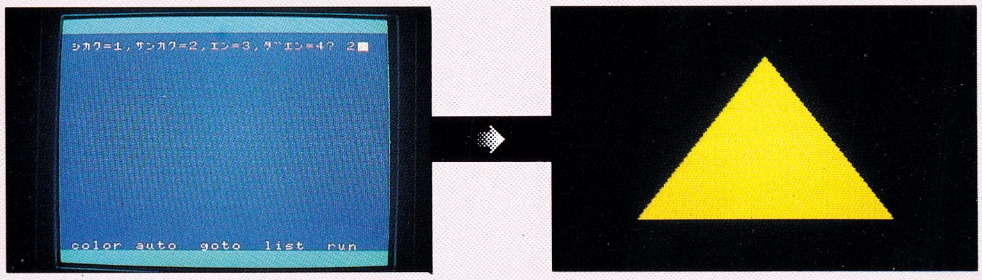
↓

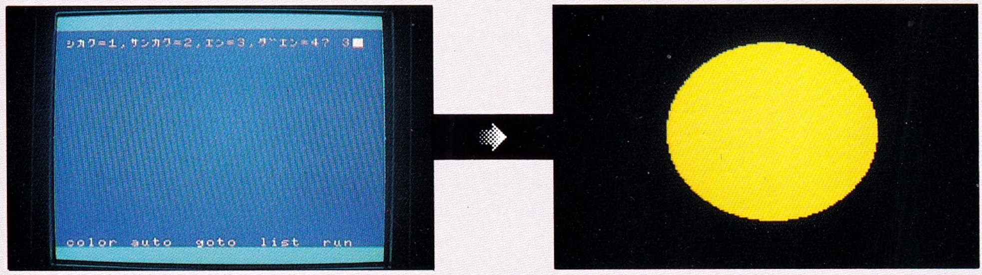


同心円状のだ円をいくつも描く(177ページ)

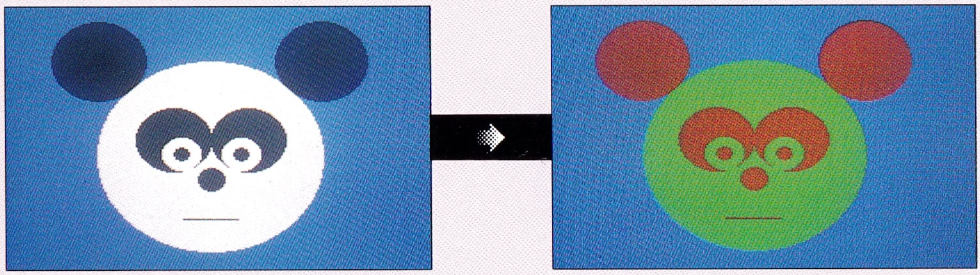


4種類の図形を指定する(179ページ)





パンダの顔の色を変える(182ページ)

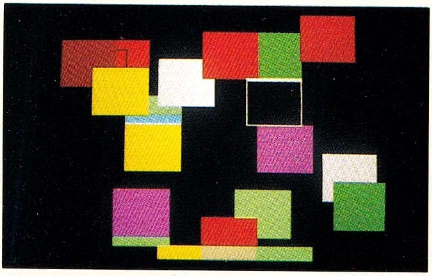


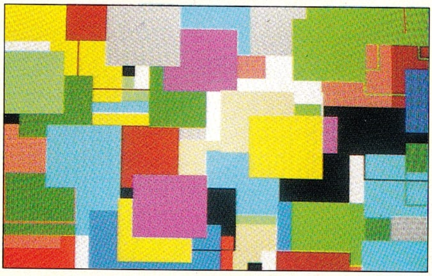
512色を12色ずつ表示する（184ページ）





さまざまな四角形を描く（191ページ）



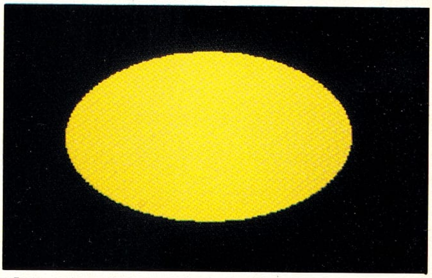


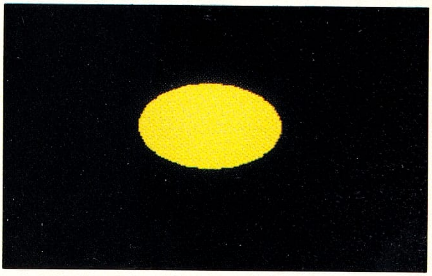
256色を一度に表示する（185ページ）



4種類の図形が拡大、縮小する〔だ円〕（189ページ）

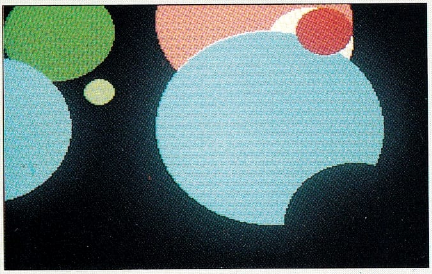


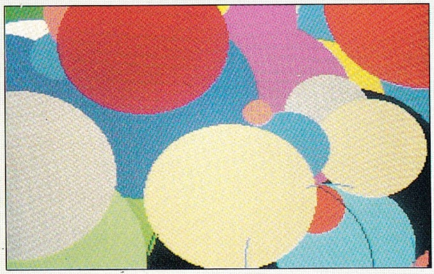




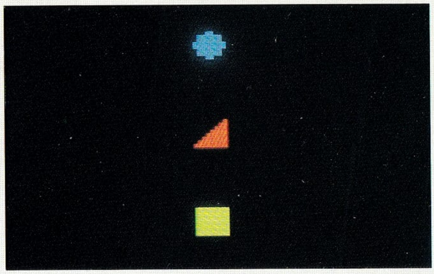
●パソコン・グラフィックスの実行結果

さまざまな円を描く(193ページ)

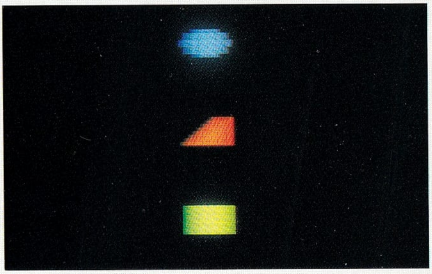




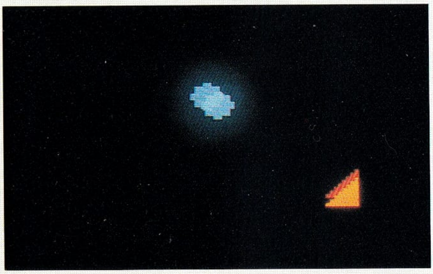
円、三角、四角をスプライト画面に表示する(199ページ)

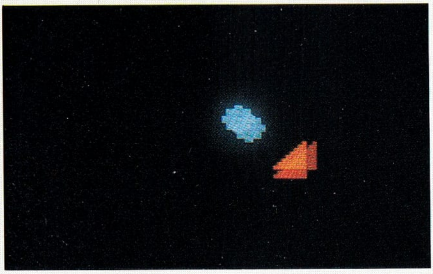


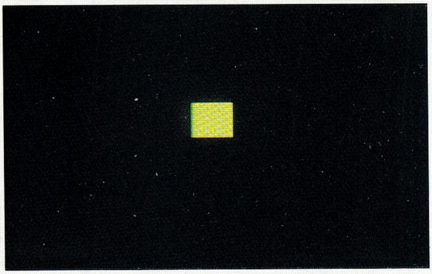
スプライト・パターンを横に動かす(201ページ)



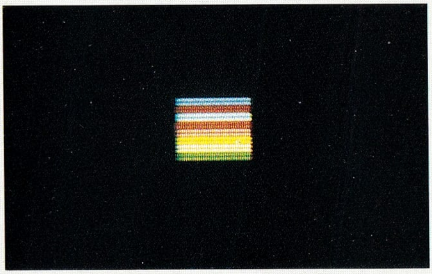
円と三角が衝突して四角になる(204ページ)







カラフルなパターンを描く(208ページ)



はじめに

MSX パソコンは、家庭用のパーソナル・コンピュータです。ホーム・パソコン、ホーム・コンピュータなどとも呼ばれるように、家庭の中で使われることを目的としてつくられたパソコンです。ということは、家族全員が使えるようなパソコンだということを意味しています。ゲームや学習に使えるばかりでなく、ママが家計簿がわりに使ったり、パパが書類作成に利用したりと、きわめて広範囲な使用法が可能です。もっと将来のことを考えれば、MSX パソコンが1人に1台の時代がくるかもしれません。

そして、家庭に入った MSX パソコンは、個々人の便利のために使われる、という以上に大きな役割を担(にな)うようになると考えられます。各家庭のパソコンを、銀行やデパートなどの大型コンピュータと回線で結べば、家にいながらパソコンを操作することによって、預金や買い物ができるようになります。家庭どうしをつなげば、電話よりもっと多くの複雑な情報をやりとりできるようになります。つまり、MSX パソコンを使うことは、未来のことにまでつながっているのです。

さて、MSX パソコンの最大の特長は、機種が違っても基本的な構造は同じだということです。ですから、どれか一社の MSX パソコンに習熟すれば、どのパソコンでも同じように使えます。また、MSX マークのついたソフトなら、どのソフトも、機種の違いを気にせずに使えます。自分の持っている機種用につくられたソフトしか使えなかったこれまでを考えると、比較にならないほど多種類のソフトを使えるのです。たくさんのソフトの中から、気に入ったものをじっくり選べるのが、私たちにとって、いちばんうれしいところです。

いろいろなソフトを使っていると、不満な点がいくつも見つかるはずです。「ここをこうしたら、もっとおもしろくなるのに……」などと思うことが出てくるでしょう。そうしたら、自分でプログラムをつくることに挑戦してください。プログラムをつくるのは、けっして簡単なことではありませんが、自力でプログラムを完成したとき、そのプログラムを実行して遊ぶときの喜びは、プログラムづくりの苦労など吹き飛ばしてくれるほど大きいのです。ぜひ、本書をフルに活用し

て、この喜びを味わってください。

本書は、パソコンに触れるのが、まったく初めての人でも、短期間でプログラムづくりの基本がマスターできるようにくふうしてあります。ステップ・バイ・ステップの考え方をとり入れ、各パートごとにプログラムの内容が段階的に高度になるようにしてあります。フローチャートを多用したこと、命令などを日本語に置きかえたことなども、理解する助けになることと思います。また、プログラムづくりには重要な、手順の説明をていねいにしているので、じっくりととり組んでください。

なお、本書は、すでに出版した『PC-8001 mk II・だれでもわかるパソコンゼミ』の内容を基本としたものです。よりわかりやすく、読みやすくするとともに、MSX2パソコンに合うように内容を変更し、必要と思われる部分については、新たに書き加えました。

プログラム入力の注意点

①新しいプログラムを入力するときは、必ず記憶消去指令（NEW（ニュー））をキーインし、RETURN キーを押してからにする。
②命令などは、大文字でも小文字でもかまわない。
③命令のあとは，1字ぶんの空白をとる。
④プログラムは、1行入力するごとに、必ず RETURN キーを押す。
⑤行番号指令（AUTO（オート））や機能キー（F2 など）を使うと、入力が効率よく行なえる。
⑥長いプログラムを入力するときは、途中（どこか区切りのよいところ）でカセットテープに記録しておく。そうすれば、うっかり電源を切ったりしても、最初からやりなおさなくてもよい。
⑦プログラムを実行するときは、実行指令（RUN（ラン））をキーインして RETURN キーを押す。あるいは F5 か F10 キーを押す。

●画面、プログラムの文字表示について

本書の画面、およびプログラムの文字は、英大文字とカタカナになっていますが、どちらも特別の場合以外は、英小文字、ひらがなでもかまいません。本書のとおりにするなら、CAPS LOCK キーを押してロックする（ランプがつく）と英大文字に、また、CAPS LOCK をロックした状態で かな キーをロックする（ランプがつく）とカタカナになります。

編者

●もくじ●

PART 7 パソコンの"得意技"データの並べかえ 137～164

PART 8 楽しもう色と形 パソコン・グラフィックス 165～209

〔写真〕平野時義 〔イラスト〕天野誠、タケバヤシてつ 〔構成〕神谷一博

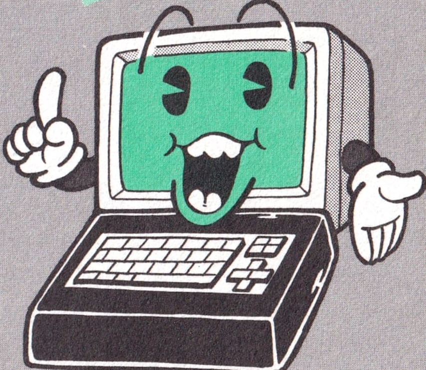
MSXという名称の由来

MSXという名称は、次のようなところからきています。最初の2文字、MとSはマイクロソフト社（Microsoft）のMとSをとったものです。最後のXは、「未知数」、あるいは「わからないもの」を意味するXです。

·PART·

1

MSX パソコンとはどういうものか



MSX パソコンの最大の特長は、MSX仕様(MSXのマークがついている)のパソコンなら、どのメーカーの機種でも基本的な部分は同じこと。このため、どの機種でも同じように使え、ソフトも共通して使えます。機種ごとに違いのあった従来のパソコンに比べて、グンと便利になったのです。しかも、これまでのMSXに加え、機能を強化したMSX 2も開発され、目的や好みに合わせて選べるようになりました。ここでは、その概要を見てみましょう。

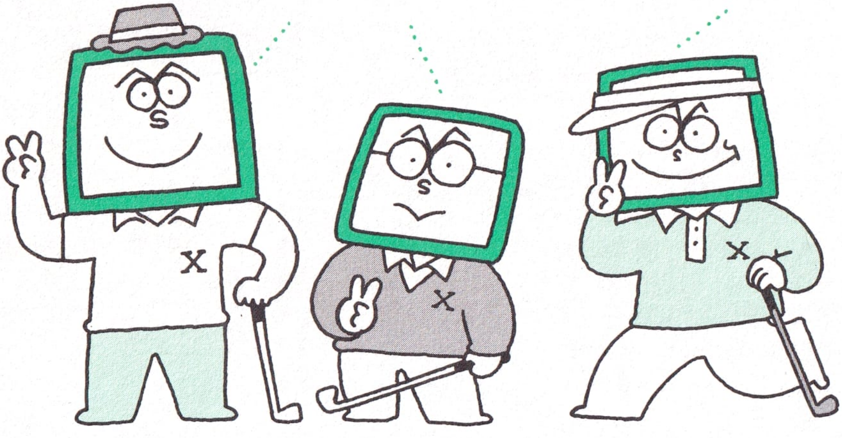
MSXとは何か

同じように使えるパソコンの規格

パソコン・ショップをのぞいてみると、各社から販売されている実にいろいろな機種が並んでいます。高いものから安いものまでさまざまですが、その中で、MSX というマークがついたものがいくつも見つかるはずです。出しているメーカーも名前も違い、デザインや値段に違いがあっても、このマークがついているものはすべて「MSX パソコン」と呼ぶことができます。

MSX とは、パソコンのハード、ソフト両面にわたる統一規格の名前です。この規格を提案したのは、アメリカのマイクロソフト社（世界有数のソフトウェア会社）と、わが国のアスキー社（パソコン関連の出版社）で、この提案に賛成したコンピュータ・メーカーや家電メーカーが、MSX 規格に基づいてつくったパソコンを販売しているのです。これらのパソコンには、すべて MSX のマークがついています。

パソコンにくわしい人ならご存じのように、これまでのパソコンは、メーカーごとに違いがありました。同じメーカーのものでも、機種が異なれば違う点がいくつもありました。このため、同じようなパソコンでも使い方が違い、いくらおもしろいソフトがあっても、機種が違えば使えませんでした。MSX パソコンは、そんな不都合をすべてとり除いてくれたのです。



MSX2はMSXの兄貴ぶん

細かくいうと、MSXパソコンには、MSXパソコンとMSX 2パソコンの2種類があります。先につくられたのはMSXで、後に機能を向上させたのがMSX 2です。MSX 2パソコンは、機能的にはMSXパソコンのお兄さんのような機種といえるでしょう。

MSXパソコンにはMSXのマーク、MSX 2パソコンにはMSX 2のマークがついています。同じマークがついていれば、どのメーカーのどの機種でも同じように使える点は共通しています。また、MSXパソコンで使えるソフトや周辺機器は、ほとんどのMSX 2パソコンで使えます。ただし、MSX 2用のものをMSXパソコンで使えるとはかぎりません。MSX 2は、MSXの機能に加えて、よりすぐれた面を持っているからです。

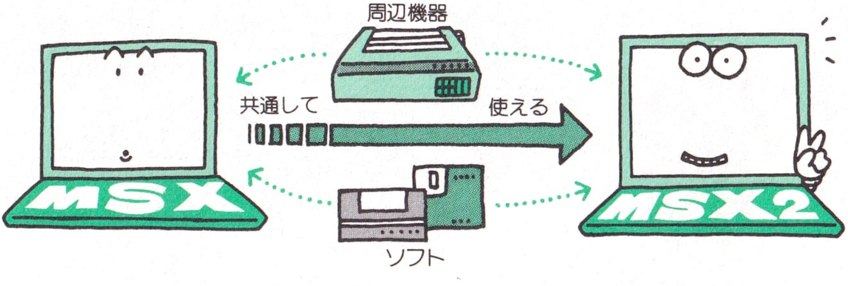
では、MSX 2は、どんな点がMSXよりすぐれているのか、次にポイントだけあげてみましょう。

●細かく鮮明な画面　MSXパソコンでは256×192ドットで構成されていた画面が、MSX 2では最高512×212ドットに。細かい絵や図形がより鮮明に描けるようになりました（くわしくはPART 8参照）。

●256色を同時に使える　使える色も、これまでの16色ではなく、256色と大幅に増え、しかも、各ドットごとに色を指定できるので、グラフィックスはグンと楽しいものになります（くわしくはPART 8参照）。

●文字は1行80字まで表示可能　MSXでは、1行に表示できる文字（アルファベット、数字、カタカナ）は40字が限度でしたが、MSX 2では最高80字まで表示できます。

●メモリ容量が64KB以上　RAMのメモリ容量が標準でも64KB（キロバイト）以上になり、大きなプログラムや多量のデータを扱えるようになりました。MSXでは、8～64KBが標準でした。



未来のMSXコンピューティング

パソコンが多くの家庭で使われるようになると、いっそう暮らしに便利に使えるようになります。具体的にどんなことができるようになるのか、例をあげてみましょう。

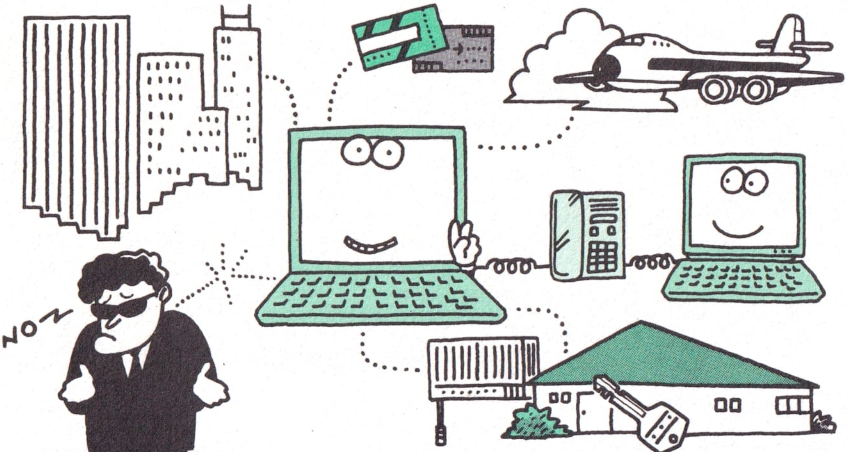
●テレコンピューティング　電話とパソコンを利用して、パソコンどうし、パソコンと大型コンピュータとのあいだで情報交換をします。ニュースなど必要な情報を大型コンピュータから自由に引き出して活用するなど、パソコンの活用範囲が大きく広がります。

●ホーム・ショッピング　画面にうつし出された商品をリストから選ぶことによって、家にいながらにして買い物ができます。飛行機や列車の予約、劇場や映画などのチケット購入もでき、代金の支払いはすべて自動的に行なえます。

●ホーム・バンキング　家庭にあるパソコンを操作することで、銀行の預金や支払いの状況を調べることができます。プリンタを使って紙に印刷すれば、預金通帳のかわりになります。

●ホーム・セキュリティ　ガス漏(も)れや漏電(ろうでん)、鍵のかけ忘れなどを、パソコンがチェックしてくれます。深夜や留守(るす)のときには、泥棒の侵入を防ぎます。

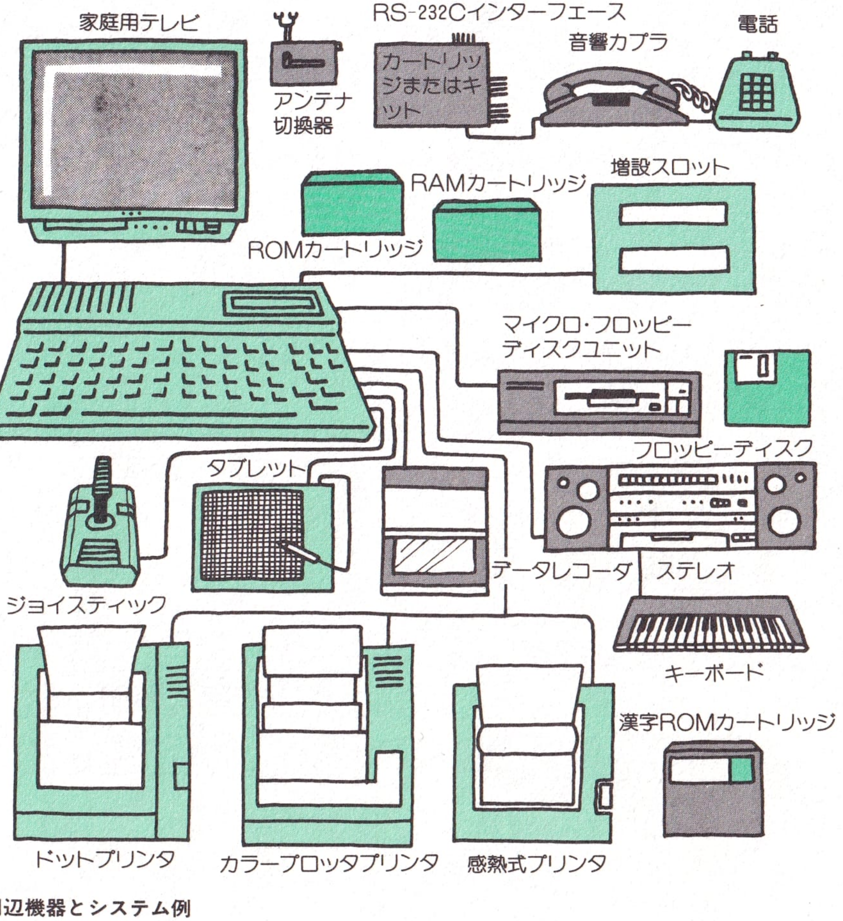
●ホーム・オートメーション　いろいろな電気製品と接続して、スイッチのon(オン)/off(オフ)などをはじめとするコントロールを自動的に行ないます。たとえば、暗くなると灯をつけ、寒ければ暖房を入れる——といったぐあいです。



MSX2の基本システム

本体を家庭用テレビにつなげば使える

MSX 2 パソコンは、家庭用のテレビに接続して使うことができます（機種によってそのまま接続できるものと、RF コンバータという装置が必要なものとがある）。市販のソフトを楽しむだけならそれでよいのですが、ほかの周辺装置をそろえれば、使い道がさらに多彩に広がります。



▲周辺機器とシステム例

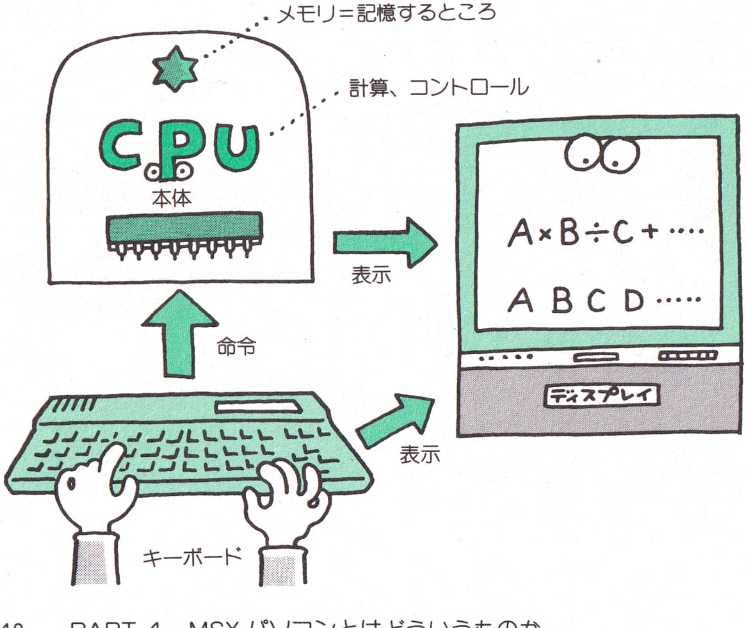
本体と周辺機器、それぞれの働きは

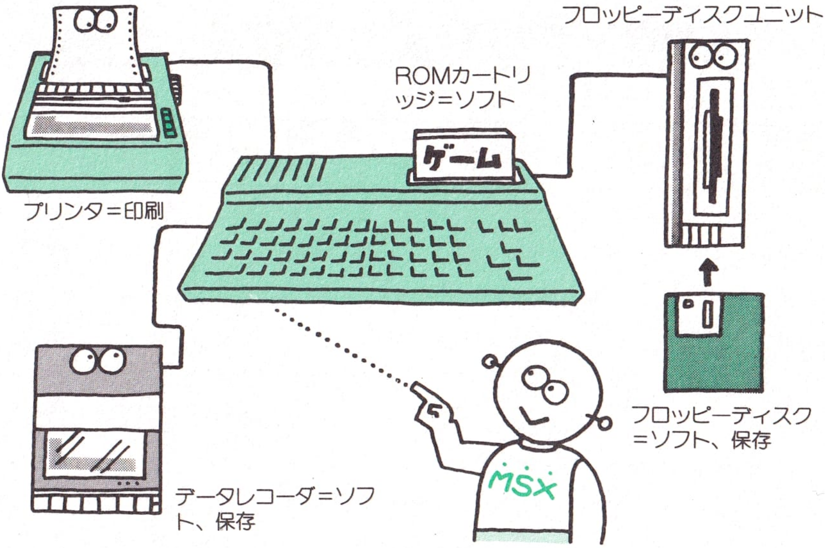
●本体はパソコンの頭脳　いろいろな計算をしたり、ものを記憶したりするのが本体の役目。全体のコントロールも行ないます。パソコンのもっとも重要な部分で、人間でいえば頭脳にあたります。くわしくいうと、計算やコントロールをする部分は中央処理装置あるいはCPU（Central Processing Unit の略）、プログラムやデータを記憶する部分はメモリと呼ばれます。

MSX パソコンも MSX 2 パソコンも、ほとんどは、本体と、次に説明するキーボードが一体となっています。

●キーボードは書きこみ装置　ボタンがいくつも並んでいるのがキーボード。ボタンのことをキーといいます。キーを操作することで、ディスプレイ画面に文字や記号を表示させたり、パソコンにいろいろなことを命令したり覚えさせたりすることができます。

●ディスプレイとは表示画面のこと　キーボードから打ちこんだ文字や数字、あるいはパソコンが処理した結果は、ディスプレイ装置に表示されます。前述したように家庭用のテレビでも使えますが、専用のディスプレイ装置を使うと、画面はより明瞭（めいりょう）で、色も鮮やかになります。





●ソフトはカートリッジかフロッピー　MSXのソフトは、電子回路をそのままパッケージしたカートリッジが中心でした。MSX 2では、このカートリッジもそのまま使えますが、ほかに、3.5インチのフロッピーディスク（略して、フロッピー）のソフトも市販されています。フロッピーは、プラスチックのカードのような外見で、中には薄い円盤が入っていて、ここにプログラムやデータを記録できるようになっています。なお、フロッピーを使うには、専用の装置（フロッピーディスクユニット）が必要です。

●保存はテープやフロッピーに　メモリに記憶させた内容は、テープレコーダを使いカセットテープに保存すると便利です。専用の装置を使うと、フロッピーにも保存が可能。こうしておけば、本体の電源を切っても内容は残ります。必要なときには、テープやフロッピーからメモリに読みこませて使います。

●印刷するにはプリンタ　計算した結果や入力したプログラム、画面上の文字や記号などを紙に印刷する装置がプリンタです。ワープロ・ソフトを使って作文や手紙を書くときには欠かせない装置です。本体に付属している機種もあります。

●RS-232Cキットでパソコン通信　電話回線を使って、パソコンどうしでデータやプログラムをやりとりするための装置がRS-232Cキットと音響カプラです。キーボードで入力した手紙をそのまま送るなどという楽しみ方ができます。なお、RS-232Cインターフェースが内蔵されているタイプの機種もあります。

MSX2パソコンのしくみ

パソコンに命令するにはパソコンのことばで

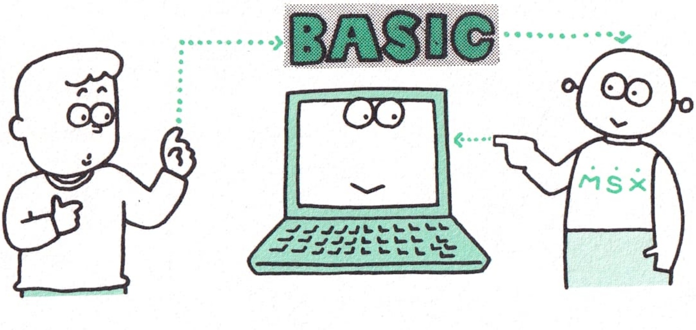
パソコンは使い方しだいで実にいろいろな仕事をこなしてくれますが、そのためには前もって、仕事の内容や手順を、人間がパソコンに教えてやらなくてはなりません。そして、人間が「実行しなさい」という命令を出してはじめて、パソコンはその仕事を始めます。

といっても、人間のことばをそのまま伝えても、パソコンには理解できません。パソコンにはパソコンにわかることばで話しかけてやる必要があるのです。このことばをコンピュータ言語と呼びます。

コンピュータ言語にはいくつかの種類がありますが、パソコンでもっとも一般的に使われているのは、ベーシック（BASIC）と呼ばれる言語です。BASIC（ベーシック）とは、Beginner's All-purpose Symbolic Instruction Code の頭文字をとった略称で、「どんな目的にも使える符号化した初心者用のことば」という意味です。“初心者用”とあるとおり簡単で覚えやすく、しかもかなり複雑な仕事までできるという特長を持っています。

同じベーシックといっても、従来のパソコンでは機種ごとに多少の違いがあったのですが、MSX パソコンのベーシックは、基本となる部分は共通ですから、どのメーカーのものでも同じように使えるわけです。

MSX パソコンのベーシックは MSX ベーシックのバージョン 1.0、MSX 2 は MSX ベーシックのバージョン 2.0 が採用されています。バージョンとは版数のことで、後者は、前者よりもグラフィックス関係の命令などが増えています。



本書で扱う命令はこれだけ

ベーシックで使う言語には、大ざっぱにいって、プログラムの中で使ってパソコンに仕事をさせる命令（ステートメント）と関数（ファンクション）、ふつうはプログラムの外で使われる制御指令（コマンド）とがあります。厳密にいえば、それぞれの働きには違いがありますが、その区別にこだわるよりも、命令、関数、制御指令のそれぞれが、どんな働きをするかを覚えるほうがたいせつです。

次に、本書で使う命令をあげておきます。この中の **COLOR SPRITE**（カラー・スプライト）は MSX 2 パソコンでしか使えません。また、**COLOR**（カラー）と **SCREEN**（スクリーン）は、命令は同じでも、その内容が MSX と MSX 2 とでは違います（本書では MSX 2 に内容を合わせた）。

◉本書で使う命令、指令、関数

書き方	読み方	名　称	働　き
END	エンド	終止命令	プログラムの実行を終了
NEW	ニュー	記憶消去指令	RAM内のプログラムを消去
AUTO	オート	行番号指令	行番号を自動的に発生させる
RUN	ラン	実行指令	入力したプログラムを実行
CONT	コンティニュー	再開指令	中断したプログラムの実行を再開する
DELETE	デリート	削除指令	プログラムを部分的に削除
LIST	リスト	リスト指令	RAMが記憶しているプログラムを画面に表示
LOCATE	ロケート	位置指定命令	文字を表示する位置を指定
PRINT	プリント	表示命令	文字や記号を画面に表示
FOR～NEXT	フォー・ネクスト	くり返し命令	FOR～NEXT 間の命令を、指定した回数だけくり返す
LET	レット	代入命令	変数に数字や文字を代入
GOTO	ゴーツー	飛びこし命令	指定した行番号に飛びこす
RND	ランダム	乱数関数	与えられた条件で乱数を発生させる
INT	インテジャー	整数関数	与えられた数値を超えない最大の整数を求める
STICK	スティック	スティック関数	ジョイスティックの傾きの方向を調べる
PLAY	プレイ	音楽命令	指定どおりに音楽を演奏する
SOUND	サウンド	音命令	指定どおりに音を出す
INPUT	インプット	入力命令	キーボードから変数に数値や文字を代入

IF～THEN～ELSE	イフ・ゼン・エルス	判断命令	条件を判断して実行する命令文を決定
BEEP	ビープ	ブザー命令	ブザーを鳴らす
READ～DATA	リード・データ	読みこみ命令	DATA のあとのデータを読みこみ、変数に割りあてる
ON～GOTO	オン・ゴーツー	ゆく先指定命令	変数で指定した行に飛びこす
RESTORE	リストア	データ指定命令	READ で読みこむ DATA 文を指定
DIM	ディメンション	配列宣言命令	配列変数の名前と要素の数を宣言
SPC	スペース	空白関数	指定した文字数だけ空白にする
MID$	ミドル・ダラー	文字列指定関数	文字列からとり出す文字の位置と長さを指定
ASC	アスキー	コード変換関数	文字をキャラクタコードに変換する
CLS	クリアスクリーン	画面消去命令	画面の表示をすべて消す
LEN	レングス	字数関数	文字の数をかぞえる
CHR$	キャラクタ・ダラー	文字変換関数	キャラクタコードを文字に変換する
VAL	バリュー	数値化関数	文字としての数字を数値に変える
SCREEN	スクリーン	モード設定命令	画面モードを決める
COLOR	カラー	色指定命令	グラフィック画面の表示色、背景色、周辺色を指定する
LINE	ライン	直線命令	指定した 2 点間に直線を引く
CIRCLE	サークル	円命令	円やだ円を描く
GOSUB～RETURN	ゴーサブ・リターン	サブルーチン呼びだし命令	サブルーチンを呼びだして実行
ON～GOSUB	オン・ゴーサブ	サブルーチン指定命令	変数で指定したサブルーチンに飛んで実行
PAINT	ペイント	彩色命令	指定した部分を指定した色でぬる
REM	リマーク	注釈命令	プログラム中に注釈文を書く
INKEY$	インキー・ダラー	入力キー関数	押されているキーの文字を関数の値とする
SPRITE$	スプライト・ダラー	動画定義変数	スプライト・パターンを定義
PUT SPRITE	プット・スプライト	動画表示命令	スプライト・パターンを表示
ON SPRITE GOSUB	オン・スプライト・ゴーサブ	動画割りこみ設定命令	スプライト・パターンの衝突による割りこみを設定する
SPRITE ON	スプライト・オン	動画割りこみ命令	スプライト・パターンの衝突による割りこみを許可する
SPRITE OFF	スプライト・オフ	動画割りこみ禁止命令	スプライト・パターンの衝突による割りこみを禁止する
COLOR SPRITE	カラー・スプライト	動画彩色命令	スプライト・パターンの色をラインごとに指定する

人間とパソコンをつなぐのは入力装置と出力装置

人間がパソコンに何かを伝えるのには、主としてキーボードを使います。ベーシックの決まりにそって適当なキーを押すことで、パソコンに命令などを伝えることができます。ほかに、ジョイスティックやライトペン（画面に当てた位置をパソコンに伝達するペン状の装置)、タブレット（座標を指定するための四角いパネル）なども、キーボードと同じような働きをします。このように、人間がパソコンに何かを伝えることを入力といい、そのための装置を入力装置といいます。

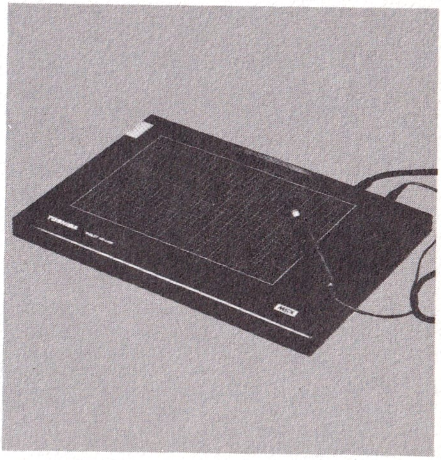
一方、パソコンがした仕事の結果や記憶していることがらを、ディスプレイ画面などに表示して人間に伝えることを、出力といいます。出力装置には、テレビなどのディスプレイ装置、プリンタ、音響装置などがあります。

コンピュータの仕事をごく単純にいえば、入力装置で伝えられたとおりに本体が仕事をし、その結果を出力装置によって伝えてくれる、ということになります。

本体は主として、CPUとメモリで構成されていることは、前述したとおりです。CPUはパソコンの頭脳の中心部で、実際にいろいろな仕事をするところですが、CPUが理解するのは機械語という、人間にはとくに意味のない数字を並べただけの言語です。もちろん、MSXパソコンは機械語で命令することもできますが、機械語は人間のことばとはあまりにもかけ離れているので、使いこなせるようになるまでがたいへんです。それで、ふつうは、より人間のことばに近いアセンブラという言語を使います。それよりさらに人間のことばに近いのがベーシックな



▲ジョイスティック



▲タブレット

```
10 DIM A(10)
20 FOR I=1 TO 10
30      INPUT A(I)
40      S=S+A(I)
50 NEXT I
60 FOR I=1 TO 9
70      PRINT A(I);"+";
80 NEXT I
90 PRINT A(10)
100 PRINT "ゴウケイハ";S
110 END
```

▲ベーシックのプログラム例

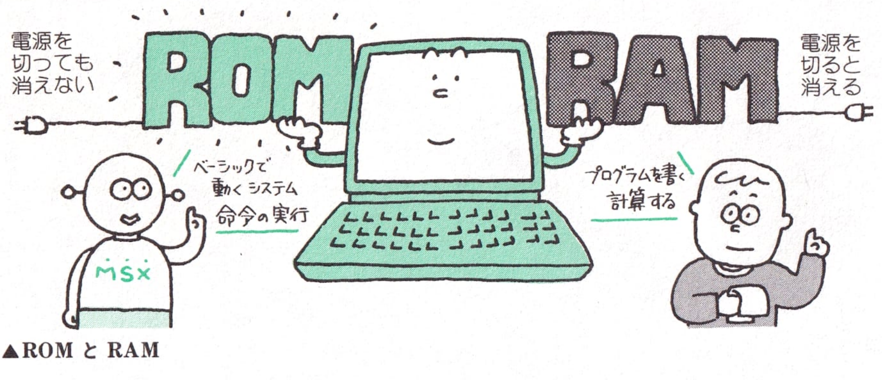
```
0C00 FF FF FF FF 7F FF 4B
0C10 4E 02 0D 0A 05 42 72
0C20 10 54 65 72 6D 69 6E
0C30 0A 0E 44 69 73 6B 20
0C40 20 4E 45 43 20 4E 2D
0C50 38 36 29 20 76 65 72
0C60 0A 0D 20 42 79 74 65
0C70 3F 52 65 64 6F 20 66
0C80 0D 0A 02 3F 20 08 1B
0C90 6B 69 70 3A 06 46 6F
0CA0 20 0D 0A 0F 55 6E 64
0CB0 6E 65 20 07 4B 57 3D
0CC0 35 36 37 38 39 41 42
```

▲機械語のプログラム例

のです。ここで必要となるのが、ベーシックを機械語に訳す通訳で、その役目を果たす機能がメモリ内に入っています。

メモリには、ROM（Read Only Memory）とRAM（Random Access Memory）の2種類があります。ROMは「読み出し専用のメモリ」ともいわれ、いつも一定の内容が記憶されていて、必要に応じ、その内容を読み出して使います。ROMの内容は、パソコンの電源が切れても消えません。ただし、ここに新しいことがらを書きこむことはできません。

一方のRAMは、「読み書き自由のメモリ」で、そこに内容を自由に書きこんだり、読み出したりできます。この点ではROMと違って便利ですが、RAMの内容は、パソコンの電源を切ると消えてしまいます。RAMの内容を保存しておくためには、カセットテープやフロッピーディスクなど、本体とは別の記憶装置が必要です。ROMとRAMをコンピュータの内部記憶装置、カセットテープなどを外部記憶装置と呼んで区別することもあります。



▲ROMとRAM

パソコンの仕事ぶりを見てみると

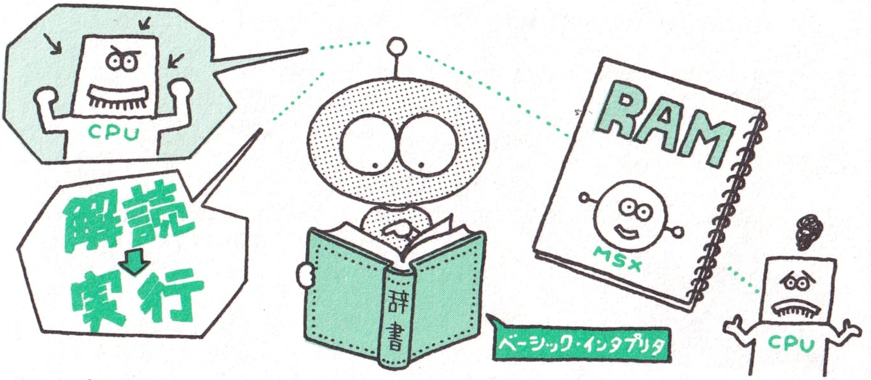
ベーシックで出した命令を受けて、パソコンがどのようなしくみで命令どおりの仕事をするのか、少しくわしく見てみましょう。

キーボードを使ってパソコンに伝えた命令は、まず、RAM(ラム)にしまわれます。このRAM(ラム)をノートと考えましょう。ノートに書かれた命令は、CPUにとってはチンプンカンプンで、なんのことかわかりません。そこで、これを読みこなすための辞書のようなものが必要になります。

CPUはその辞書を持ち出してきて、伝えられた命令が、そこにのっているかどうかを調べます。のっていなければ、エラーメッセージといって、誤りがあったことを表示します。命令が辞書にのっていれば、辞書に基づいて解読にかかります。つまり、CPUにとっては理解できないベーシックの命令を、機械語になおすのです。これでようやく、CPUに通じることばになります。

ただし、この機械語はことばの数がきわめて少なく、単純な語彙(ごい)しかありません。ですから、ベーシックの命令を機械語になおすと、単純な命令の集合となります。たとえば、「棚の本をとれ」という命令は、「立て」「棚に向かって歩け」「止まれ」「手を伸ばせ」というように、いくつかの命令に分解されてしまうのです。どんなに複雑な命令でも、適当な区切りをつけ、以上のような過程でひとつの命令を解読→実行し、終わると次の命令の解読→実行というぐあいに何度もくり返して実行していきます。

機械語の辞書や解読作業の手順、方法は、ROM(ロム)の中にすべてあらかじめ書きこまれています。これを、ベーシック・インタプリタといいます。



▲インタプリタのしくみと働き

COLUMN

◉プログラムを保存しよう◉

いろいろ苦労してつくったプログラムも、電源を切ると一瞬にして消えてしまう。また、長いプログラムを入力している最中に、ついうっかり電源を切ってしまうと、全部パァになる。

これでは不便このうえない。プログラムは、ぜひカセットテープレコーダに保存しておこう。長いプログラムを入力しているときは、区切りのいいところでテープに保存しながら入力していくといい。

カセットテープレコーダは、パソコン専用のものがある。これを使うにこしたことはないけれど、なーに一般のもので十分まにあう。

プログラムをカセットテープに記録することをセーブ、逆に、テープのプログラムを本体に送りこむことをロードという。一度セーブしたプログラムは、何度でもロードすることができる。扱い方は、音楽テープと同じだと思ってよい。

①本体にテープレコーダを接続する（マニュアル参照）

セーブ

②テープレコーダを録音状態にする

押す → REC　PLAY

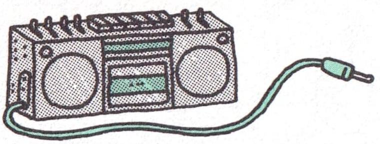
③キーボードからセーブの命令を入力

CSAVE "GAME 1" RETURN

CSAVE ← セーブの命令　"GAME 1" ← 6字以内のタイトル（自分でつける）

④テープがまわりはじめ、画面が下のようになれば完了

```
CSAVE "GAME 1"
Ok
■
```



ロード

②テープレコーダを再生状態にする

押す → PLAY　（REC　PLAY）

③キーボードからロードの命令を入力

CLOAD "GAME 1" RETURN

CLOAD ← ロードの命令

④画面が下のようになれば完了

```
CLOAD "GAME 1"
Found : GAME 1
Ok
■
```

「GAME　1」の前にほかのプログラムが入っていて、それを読みとばした場合は、「SKIP : GAME　A」などという表示が出る。「SKIP」は、読みとばしたことを意味し、コロン（:）のあとに続くのは、読みとばされたプログラムの名前。

PART 2





セッティングとキーインの基本

MSX 2パソコンのシステムは、ひととおりそろいましたか？では、各装置を適当な場所に設置して接続しましょう。これをセッティングといいます。ここでミスすると、パソコンは動いてくれないので慎重に。キー操作は慣れるにつれて速くなります。基本をよく頭に入れて、パソコンに触れる機会をなるべく数多くつくることが、上達のポイントです。



設置から接続までのポイント

こんな場所に置いてはいけない

コンピュータは、精密な電子部品からできています。設置場所にも、注意が必要です。次にあげるような場所に置くことは、故障などを起こす原因になるので避けてください。

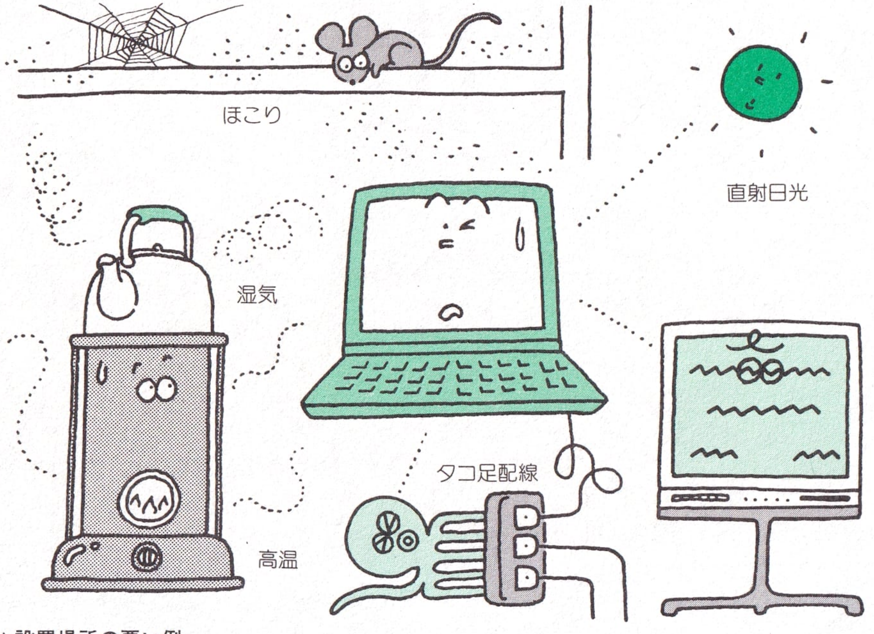
①風通しの悪い場所や、極端に低温あるいは高温になる場所。また、温度変化の激しい場所。

②直射日光が当たる場所。暖房器具など高熱を出す器具の近く。

③極端に湿気の多い場所、ほこりの多い場所。

④ラジオ、テレビなどのそば（ラジオやテレビに雑音が入る）。

また、通風孔をふさいだり、上に重いものを置いたりしないという注意も守ってください。もうひとつ、タコ足配線の末端を電源として使うことも避けてください。電圧が不安定になり、思わぬトラブルにつながったりします。



▲設置場所の悪い例

接続するときの注意点

本体とほかの装置を接続する部分を端子といい、本体の背面や側面にあります。それぞれの装置と適切な端子をケーブル（コード）でつなげばよいのですが、機種によって細かい違いがあるので注意してください。たとえば、本体と家庭用テレビをつなぐ場合に、そのままケーブルでつなげるものと、RF コンバータという装置が必要なものとがあります。取扱説明書をよく読んで、正しく接続しましょう。

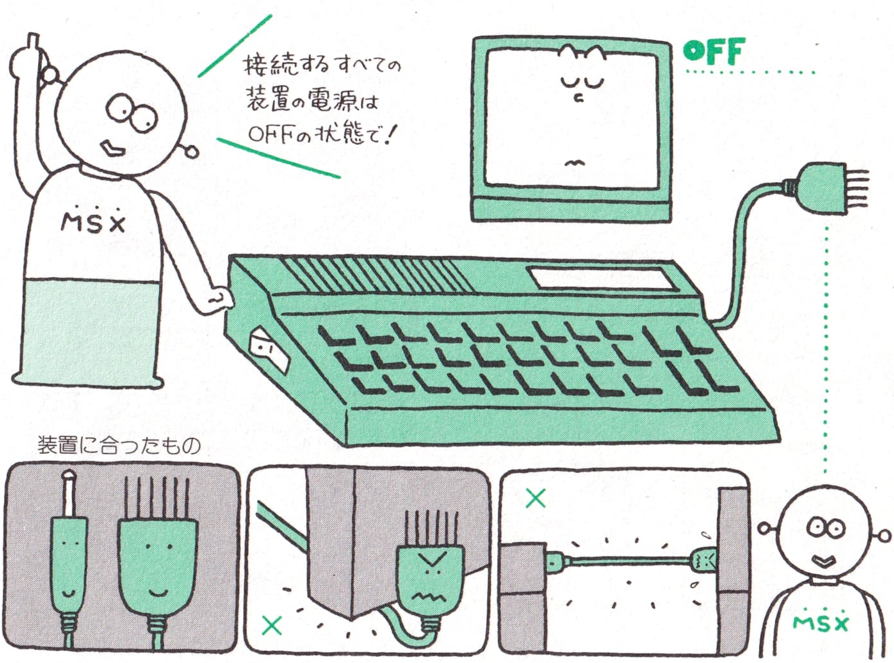
次にあげる点は、どの機種にも共通にあてはまります。必ず守ってください。

①接続するすべての装置の電源を切った（off）状態で接続します。切り離すときにも同様です。

②ケーブルは、それぞれの装置に合ったものを使います。ケーブルの形や接続部をよく調べて接続します。

③ケーブルが正しい向きになっているかどうか、たしかめましょう。

④ケーブルがピーンと張りつめたり、ケーブルの上に物が置いてあったりしないように、各装置の位置を調整します。



カートリッジの接続は電源をオフにして

MSX 2 パソコンには、本体(キーボード) の上面や横に細長い穴が 1、2 個ついています。この穴はスロットといい、プログラムが入っているカートリッジを差しこむためのものです。

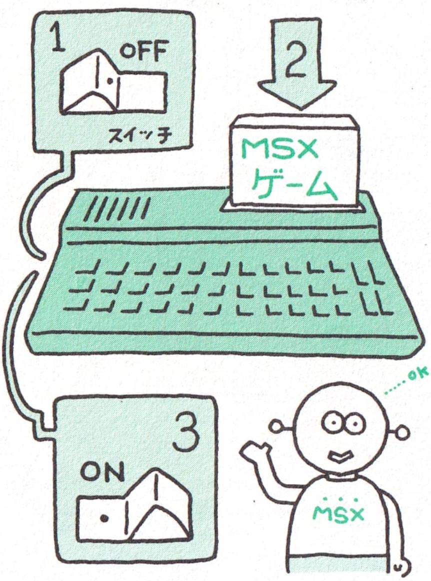
カートリッジは、外見はカセットテープに似ていますが、中身はテープではなく電子回路です。これをスロットに差しこむことによって、ゲームはもちろんのこと、文章を作成・印刷したり、家計簿をつけたりするのに、パソコンを使うことができるようになります。

カートリッジには多くの種類がありますが、MSX か MSX 2 のマークが入っているものであれば、ほとんどの機種に使えます。

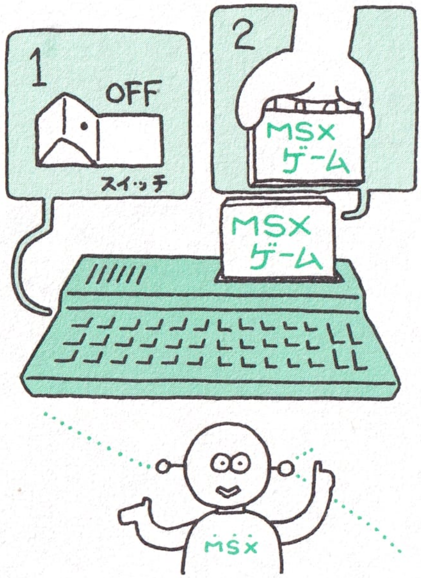
カートリッジを接続する場合には、次の点を守ってください。

①本体に記憶されているプログラムなどの内容は、接続前にカセットテープに記録しておく（カートリッジを接続すると、プログラムなどはすべて消えてしまいます)。

②本体の電源を切って（off（オフ））から接続する（電源を入れたままだと故障の原因になることがあります)。



▲カートリッジの接続



▲カートリッジのとりはずし

ではスイッチを入れます

スイッチ・オンは周辺装置から

電源スイッチを入れる順序は、「まず周辺装置、最後に本体」が原則です。逆にすると、思わぬ事故の原因となることがあるので注意します。たとえば、テレビ→カセットテープレコーダ→プリンタ→本体、という順序で電源を入れます。

機種によっては、本体についているチャンネル切りかえスイッチと家庭用テレビのチャンネルを、その地方で使われていないチャンネル（東京近辺なら2チャンネル、大阪近辺では1チャンネル）にしておかなくてはならないものもあります。

本体の電源スイッチを入れると、キーボードのはしにある電源ランプがつき、テレビ画面にまず「MSX」などという表示が出て、数秒してから下のような文字があらわれます。このような画面になれば、これ以後、ベーシックの決まりに基づいてパソコンに命令をすることができます。この画面のことを初期画面といいます。

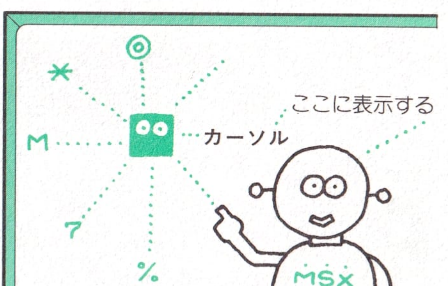
上から3行めまでの英語は、「1985年にマイクロソフト社によってつくられたMSXベーシックの第2.0版で、28815文字ぶん（記憶容量により違う）使えます」という意味です。その下の「Ok」は、命令待ちの状態であることを示しています。

「Ok」の下で点滅している「■」を、カーソルといいます。キーボードから打ちこんだ文字や記号は、このカーソルのある位置に表示されます。

画面のいちばん下に表示されている行は、1～5の機能キー（32ページ参照）にどんな文字が割りあてられているかをあらわしています。この行は、**KEY OFF**（キー オフ）という命令を入力しないかぎり、いつも表示されています。



▲初期画面



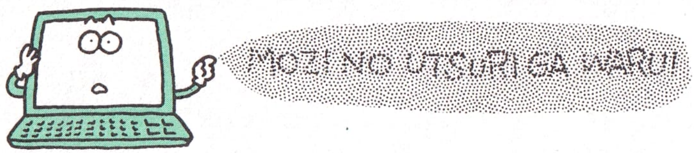
▲カーソルの働き

初期画面のようすがおかしいときは

初期画面が前ページのように正しく表示されない場合は、それぞれのケースに応じて、次のようなことを行なってみます。

画面に何も表示されない

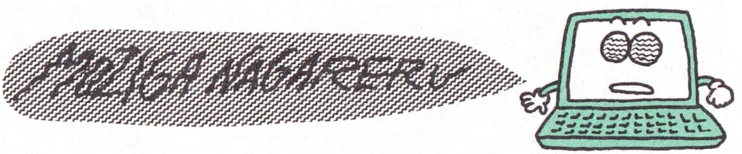
- ●本体とテレビの電源プラグがコンセントに差しこまれているかどうかを調べる
- ●本体とテレビの電源ランプがついているかどうかを調べる
- ●本体とテレビが正しく接続されているかどうかを調べる（接続のぐあいや端子の位置など）
- ●本体のチャンネル切りかえスイッチとテレビのチャンネルが一致しているかどうかを調べる（機種による）
- ●一度電源を切って、5～6秒後に再び電源を入れてみる



文字のうつりが悪い

- ●テレビのコントラストつまみを調整してみる
- ●テレビの明るさ（ブライト）つまみを調整してみる

(注)テレビ画面の色やコントラスト、それに音の大きさは、テレビ放送の映像とは違います。カラーグラフィックスや音楽演奏を楽しむ場合には、適切に調整することが必要です。



文字が流れる

- ●文字が縦に流れる場合には、テレビの垂直同期のつまみを調整してみる
- ●文字が横に流れる場合には、テレビの水平同期のつまみを調整してみる

まず覚えよう、キー操作の基本

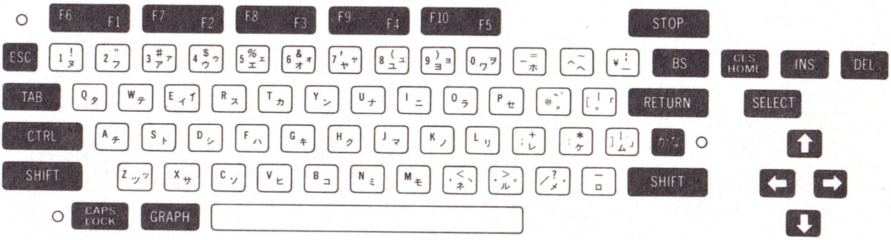
キーは軽く押してすぐにはなすのが基本

キーボードをじっくりながめてください。機種によって多少の違いはありますが、基本はだいたい同じです。キーの色によって全体が2種類に大別されていることに気づきましたか。一方が一般キー（文字キー）、他方が特殊キーです。

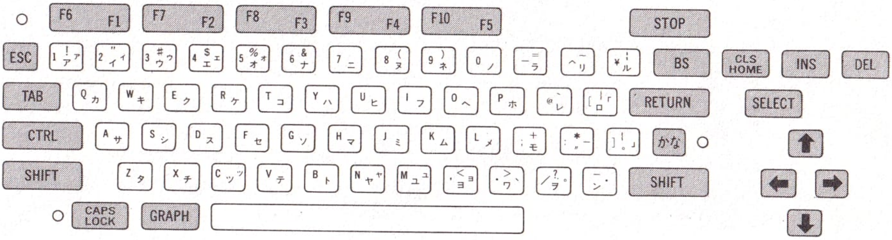
一般キーのうち、どれかすきなキーを押してみましょう。ひとつのキーに2文字以上書かれたものもありますが、どのキーでも、そのうちの1文字だけがテレビ画面に出てきます。同じキーを押し続けていると、そのあいだは同じ文字が連続して表示されます。この働きを、オートリピート機能といいます。1文字だけ表示したいときには、キーを軽くポンと押して、すぐにはなすことです。

アルファベットの並び方はどの機種も共通ですが、カタカナ（または、ひらがな）の並び方には注意してください。一見規則的でないもの（JIS規格配列）と、横に50音順に並んでいるものとがあります。

いちばん下にある細長い、表面に何も書いていないキーは空白キー（スペース・キー）といい、1文字ぶんの空白をつくります。



▲JIS規格配列のキーボード



▲50音配列のキーボード

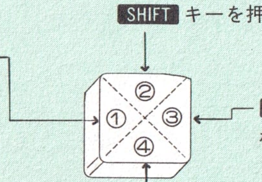
キーの文字は特殊キーとの組み合わせで表示

数字のキーを押すと数字、アルファベットのキーを押すと、そのアルファベットの小文字がディスプレイに表示されることがわかったでしょう。これら以外の、アルファベットの大文字やかな、またキーの上部や右の部分に書いてある記号などを表示するには、特殊キーと組み合わせる必要があります。ここで使う特殊キーは、SHIFT、かな、CAPS LOCK（または CAPS）の3つです。

このうちかなキーとCAPS LOCKキーは、一度押すと押されたままの状態になり、手をはなしてももとに戻りません。これを、「ロックする」といいます。ロックしたキーをもう一度押すと、押さえる前の状態に戻ってロックが解除されます。この2種類のキーには、すぐ横かキーの中に小さなランプがついています。このランプはロック状態のときに点灯し、ロックを解除すると消えます。キー操作のときには、このランプを確認してください。

SHIFTキーは、一度押してもすぐもとに戻り、ロックした状態になりません。使うときには、このキーを押しながら、適当な一般キーを押します。左右2つありますが、どちらを使ってもかまいません。

電源を入れたあと、キーに書いてある文字や記号のうち、希望するものを打ちこむには、それが書いてある位置によって次のようになります。ただし、機種によっては若干の相違があるので注意してください。



〈例〉



- そのまま押す→英語小文字「e」
- CAPS LOCK を押してから→英語大文字「E」
- かな キーを押してから→ひらがな「い」
- CAPS LOCK キーと かな キーを押してから→カタカナ「イ」
- かな キーを押してから SHIFT キーを押しながら→ひらがな小文字「ぃ」
- CAPS LOCK と かな キーを押してから SHIFT キーを押しながら→カタカナ小文字「ィ」
- SHIFT キーを押しながら→何も表示されない

▲一般キーの表示方法

特殊キーの機能はこれだけある

●特殊キーの働き

SHIFT キー、かな キー、CAPS LOCK キー以外の特殊キーは、それぞれ次のような働きをします。GRAPH（GRPH）キー、F1 ～ F10 キーについては次項で説明します。

同じ働きをするキーでも、機種によって文字や記号が違う場合があります。下の表では※で示してあります。

なお、すでにおわかりと思いますが、特殊キーを押しても、そこに書いてある文字は画面に表示されません。特殊キーに書いてある文字は、一般キーと違って、キーの働きをあらわしていることに注意しましょう。

◉特殊キーの種類と機能

（注）　※印は機種による違い。働きは同じ。

キー	名　　称 （カッコ内は一般名）	機　　　能
RETURN ※ ENTER ※ ↵	入力終了キー （リターン・キー エンター・キー）	プログラムの1行ぶんや、データを入力し終えたときに、このキーを押してコンピュータに入力終了を知らせる
ESC	拡張キー （エスケープ・キー）	コンピュータどうしを接続するとき利用 通常は使わない
TAB	タブ・キー	押すと、カーソルが8文字単位で右へ移動する その間は空白になる
CTRL	組み合わせキー （コントロール・キー）	一般キーと組み合わせて使い、特殊キーと同じような機能を行なう（別表参照）
STOP	停止キー （ストップ・キー）	①押すと、プログラムや命令の実行を一時停止する。もう一度押すと、実行が再開する。一時停止では、キーによる入力はできない ② CTRL を押しながらこのキーを押すと、実行は中断し、「Ok」表示とカーソルが出る。この場合は、キーによる入力ができる
CLS HOME ※ HOME CLS ※ HOME	画面消去キー （ホームクリア・キー）	①押すと、カーソルが画面の左上のはしに移動する ② SHIFT キーを押しながらこのキーを押すと、カーソルが画面左上端に移動すると同時に、画面上の表示がすべて消える
← → ↑ ↓ ※ ＜ ＞ ∧ ∨	カーソル移動キー	カーソルを矢印の方向へ移動する

INS	挿入キー (インサート・キー)	押すと、文字が挿入できる状態（挿入モード）になる。この状態でキーを押すと、文字がカーソルの位置に挿入される。この状態を解除するには、このキー、あるいは入力終了キーやカーソル移動キーを押す
DEL	削除キー (デリート・キー)	押すと、カーソルの位置の文字が消え、カーソルの右側にある文字が左へ1文字ぶん移動する
BS ※ BACK SPACE	左削除キー (バックスペース・キー)	カーソルの直前、左側の1文字が消え、カーソルの位置にある文字が左へ1文字ぶん移動する 削除キーとの違いに注意
SELECT	とり消しキー (セレクト・キー)	まちがっているデータを無視する 通常は使わない

◉組み合わせキーの使い方

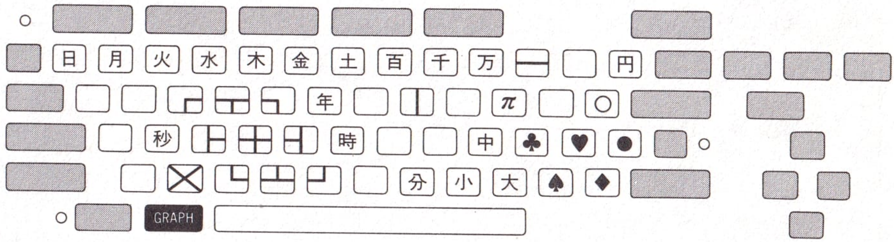
（注）ほとんどは、特殊キーの働きと同じ。

CTRL を押しながら	B	カーソルを前の項目の先頭に移動
	C	INPUT 文などの実行を停止する（CTRL＋STOP）と同じ）
	E	カーソルがある位置より後ろの行を削除する
	F	カーソルを次の項目の先頭に移動する
	G	ブザーの音を出す
	H	カーソルの左側の1文字を消す（BSと同じ）
	I	カーソルが8文字単位で移動する（TABと同じ）
	J	カーソルを下に移動（改行）する
	K	カーソルを画面左上端に移動する（CLS HOMEと同じ）
	L	画面全体の表示を消し、カーソルを画面左上端に移動する （SHIFT＋CLS HOMEと同じ）
	M	入力終了を知らせる（RETURNと同じ）
	N	カーソルを行の最後に移動する
	R	文字を挿入する（INSと同じ）
	U	カーソルのある位置の1行を消す
	¥	カーソルを右へ移動する（→と同じ）
	ム	カーソルを左へ移動する（←と同じ）
	へ	カーソルを上へ移動する（↑と同じ）
	SHIFT と ロ	カーソルを下へ移動する（↓と同じ）

✣GRAPHを使えば記号も漢字も打ちこめる

空白キーのすぐ横にあるGRAPHは、グラフィック・キー（またはグラフ・キー）といいます。このキーを押しながら別のキーを押すことで、さまざまな記号や線、漢字を打ちこむことができます。これらの記号や漢字をまとめて、グラフィック・シンボルといいます。

機種によって、グラフィック・シンボルがキーに書かれているものと、書かれていないものがあります。後者の場合は、下の図を参考にして付属のシールをキーにはり、目的のグラフィック・シンボルの位置がわかるようにします。



▲グラフィック・シンボル

✣あらかじめ文字がセットされた機能キー

キーボードのいちばん上の列には、同じようなキーが並んでいます。F6/F1 F7/F2 F8/F3 F9/F4 F10/F5の5つで、これらを機能キー（ファンクション・キー）といいます。上下2列に書いてあるのは、そのまま押す場合と、SHIFTキーを押しながら押す場合とで違うからです。たとえば、F6/F1をそのまま押せばF1、SHIFTキーを押しながら押せばF6になります。結局、機能キーは、F1～F10までの10種類あることになります。

機能キーには、あらかじめそれぞれに文字がセットされています。これらの文字は、ひとつの命令になっているので、プログラムを入力するときなどに便利です。たとえば、F3には“goto”（ゴーツー）という文字がセットされています。これは、あとでくわしく説明するように、「飛びこし命令」というよく使う命令です。わざわざ**g**、**o**、**t**、**o**、と4つのアルファベットを打ちこまなくても、F3をポンと押すだけですみます。

機能キーにどのような文字（命令）がセットされているかは、以下の表を参考にしてください。F1～F5までは、画面のいちばん下のところに表示されます。また、機能キーには、自分ですきな文字をセットすることができます。パソコンにさせる仕事の内容に合わせて、よく使う文字をセットしておくと便利です（76ページ参照）。

●機能キーとセット内容

区分	押　す　キ　ー	セット文字	命令の内容
F 1	F6/F1	color ␣	色を指定する
F 2	F7/F2	auto ␣	行番号を表示
F 3	F8/F3	goto ␣	指定の行へ飛ぶ
F 4	F9/F4	list ␣	プログラム・リストを表示する
F 5	F10/F5	run CR	実行する
F 6	SHIFT を押しながら F6/F1	color 15, 4, 7 CR	画面の色を最初の状態にする
F 7	SHIFT を押しながら F7/F2	cload"	テープのプログラムを読みこむ
F 8	SHIFT を押しながら F8/F3	cont CR	実行を継続する
F 9	SHIFT を押しながら F9/F4	list. CR ↑ ↑	プログラムの特定の1行を表示
F 10	SHIFT を押しながら F10/F5	cls run CR	一度、前の画面を消してから実行する

(注) ␣は1字ぶんの空白のこと。
CRは、入力終了キー（RETURN）が押されたのと同じ働き。
↑は、↑が1回押されたのと同じ働き。
clsは、画面消去キー（CLS HOME）が押されたのと同じ働き。

文字の修正法に慣れよう

訂正、削除はこの方法で

実際にMSXパソコンを使う場合、キーの押しまちがいなどのために、修正しなくてはならないことが起こります。文字の修正は、おもに、カーソル移動キー←→↑↓と、削除キーDEL、挿入キーINSを使って行ないます。

なお、文字の訂正、削除、あるいは挿入、追加（次項参照）などが終わったあとは、必ず入力終了キーであるRETURNを押してください。このキーを押さないと、画面上では修正されていても、パソコン内部までそれが伝わらないので、結局、何も修正されていないことになります。RETURNを押すと、カーソルが次の行の先頭に移動しますから、それを確認するくせをつけておくと、失敗せずにすみます。

以下の入力例でRETURNを押すと、ビーッと音がして「Syntax error（シンタックス エラー）」という文字が出ることもありますが、気にせずに進めてください。

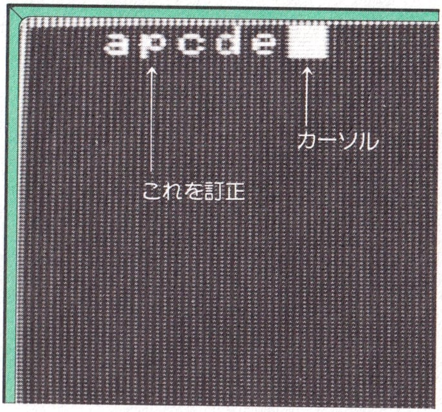
✣訂正は上から正しい字を打つ

①カーソル移動キーを使って、カーソルを訂正する文字まで移動する。

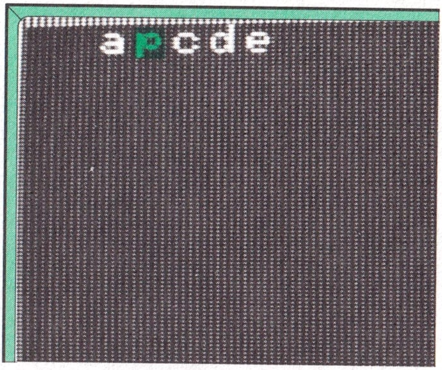
②正しい文字を打ちこむ。

③RETURNを押す。

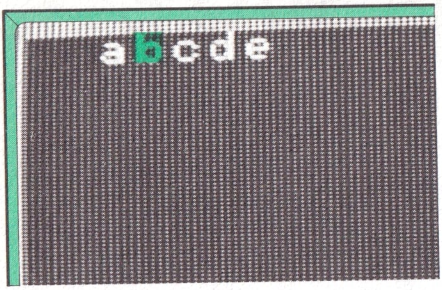
例 pの文字をbに変える



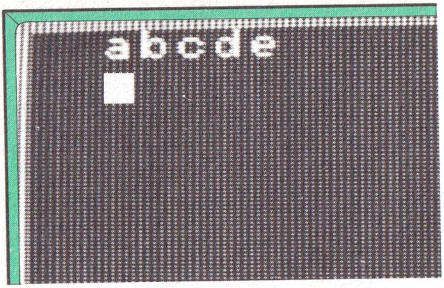
①カーソルをpのところまで移動



②[B]キーを押す



③ RETURN を押す



✣追加は最後に打ちこむ

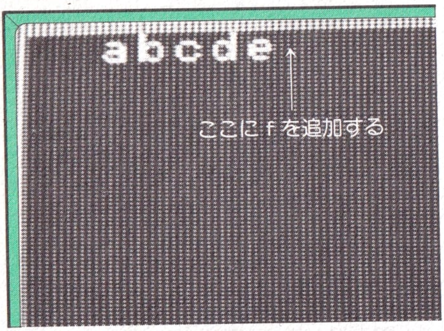
追加はもっと簡単にできます。

①カーソル移動キーで追加する文字の位置に、カーソルを移動する。

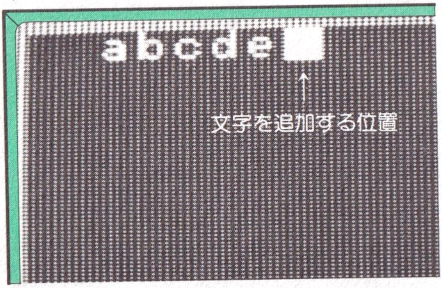
②追加する文字を、キーインする。

③ RETURN を押す。

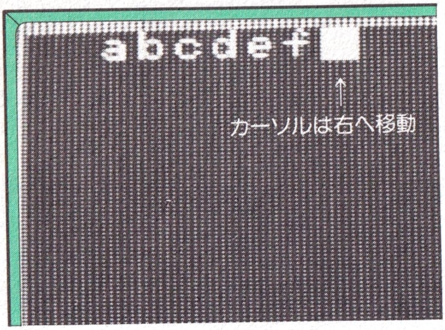
例 eのあとにfを追加する



①カーソルをeの右に移動



②[F]キーを押す



③ RETURN を押す



DEL キーで削除は簡単にできる

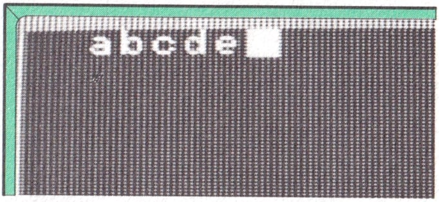
よぶんな文字を削る場合です。

①カーソル移動キーで削除する文字のところにカーソルを合わせる。

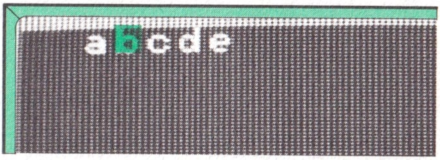
② DEL を押す。

③ RETURN を押す。

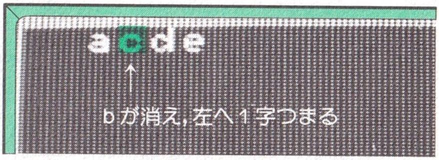
例 b c d を削除する



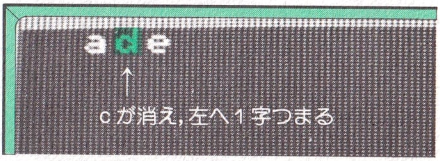
①カーソルを削除する文字に合わせる



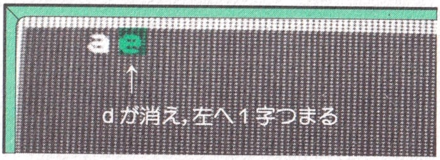
② DEL を１回押す



③もう一度 DEL を押す



④もう一度 DEL を押す



⑤ RETURN を押す



DEL キーは、１回１回押すほうが確実です。オートリピート機能を利用すると、連続して文字を削除できて便利ですが、キーから指をはなすタイミングを誤ると、必要な文字まで削除してしまうことがあります。

練習問題 1

画面に「アオウエカエオ」と表示し、次のような順序で「アイウエオ」という表示に修正してみよう。

①「オ」を「イ」に訂正して「アイウエカエオ」とする。

②２文字ある「エ」のうち、どちらかの「エ」を削除する。

③「カ」を削除する。

挿入はカーソル移動と INS キーで

打ち忘れた文字をあとから入れる場合には、次のようにします。

①カーソル移動キーで、新しい文字を挿入する位置にカーソルを移動する。

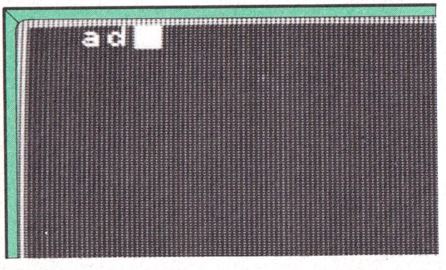
② INS を押す（機種によっては、ここでカーソルの形が変わる）。

③挿入する文字を順にキーで打ちこむ。

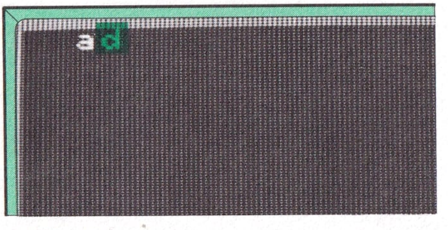
④ RETURN キーを押す。

この際、同じ行で挿入以外に削除や追加、訂正などを行なう場合には、挿入が終わったあとで、もう一度 INS を押すか、カーソルをカーソル移動キーで移動させてから、ほかの作業にとりかかってください。それを忘れると、いつまでも挿入しかできないことになります。

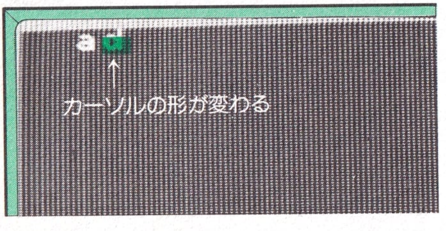
例 aとdのあいだにbとcを入れる



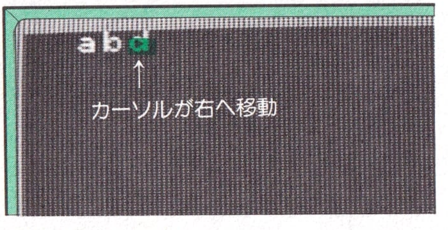
①カーソルを挿入位置に移動



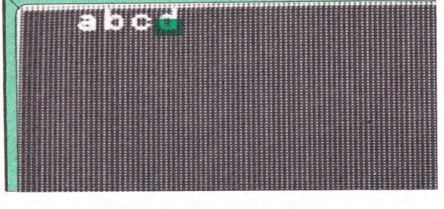
② INS を押す



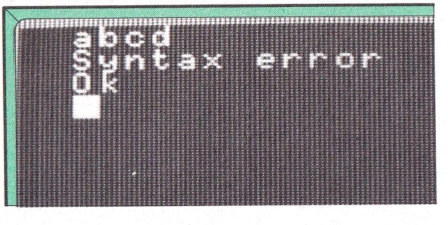
③ B を押す



④ C を押す



⑤ RETURN を押す



まちがえやすいキー操作

キーの文字や記号の中には、形が似ていて、混同しやすいものがあります。画面上ではあまり違わなくても、思わぬミスの原因となるので注意しましょう。次に、まちがえやすいキーをあげておきます。

①マイナス、アンダーライン、長音記号⇨

マイナス記号	(－)	[- ホ]	(例) 5 － 3 ⇨ [5] [- ホ] [3]
アンダーライン	(_)	[_ ロ]	(例) A_ ⇨ [A] [_ ロ]
かなの長音記号	(ー)	[¥ ー]	(例) ローマ ⇨ [かな] [ロ] [¥ ー] [マ]

②コンマ、かなの読点、アポストロフィ、アクセント記号

コンマ	(，)	[, ネ]	(例) a,b ⇨ [A] [, ネ] [B]
かなの読点	(、)	[, ネ]	(例) フネ、ウシ ⇨ [かな] [フ] [ネ] [SHIFT] [, ネ] [ウ] [シ]
アポストロフィ	(’)	[' ヤ]	(例) 1'game ⇨ [1] [SHIFT] [' ヤ] [G] [A] [M] [E]
アクセント記号	(\`)	[\` ゛]	(例) tomàto ⇨ [T] [O] [M] [SHIFT] [\` ゛] [A] [T] [O]

③ピリオド、かなの区切り記号

ピリオド	(．)	[. ル]	(例) end. ⇨ [E] [N] [D] [. ル]
かなの区切り記号	(・)	[/ ・]	(例) コマ・マリ ⇨ [かな] [コ] [マ] [SHIFT] [/ ・] [マ] [リ]

④半濁点、かなの句点（マル）

半濁点	(゜)	[゜]	(例) パイプ ⇨ [かな] [ハ] [゜] [イ] [フ] [゜]
かなの句点	(。)	[. ル]	(例) カイタ。 ⇨ [かな] [カ] [イ] [タ] [SHIFT] [. ル]

⑤ダブルコーテーション、濁点

ダブルコーテーション	(")	[2 "]	(例) print"a" ⇨ [P] [R] [I] [N] [T] [SHIFT] [2 "] [A] [SHIFT] [2 "]
濁点	(゛)	[゛]	(例) カバ ⇨ [かな] [カ] [ハ] [゛]

(注) 上にあげたキーは標準的なキーボードの一例です。機種によっては多少の違いがあるので、説明書で確認してください。

練習問題 2

画面に「パソコン」と表示し、次のような手順で「パーソナルコンピュータ」と修正してください。

①「パソ」を「パーソ」となおし、「コン」とのあいだに「ナル」を挿入する。

②「コン」のあとに「ピュータ」を追加する。

COLUMN

◉ひらがな・カタカナをローマ字で入力◉

MSX2パソコンでは、ひらがなやカタカナをローマ字で打ちこんで変換して表示できるようになった。たとえば「あさ」と表示したいなら、ASAと打ちこむわけ。これで画面には「あさ」と出る。

これが、ローマ字入力。アルファベット26文字のキーだけを使うので（実際に使うのはもっと少ない）、ひらがな（またはカタカナ）のキーを探すよりもずっと速いことが利点だ。ただ、単純にいえば、キーを押す回数が2倍になる（「さ」ならSA）。

ローマ字入力ができるようにするには、SHIFTキーを押しながらかなキーを押す。キーのランプがつき、この状態でローマ字を打ちこむと、ひらがなに変換される。カタカナは、ひらがなの状態にしたうえで、さらにCAPS LOCKキーを押す。ランプがつき、これでローマ字がカタカナになる。

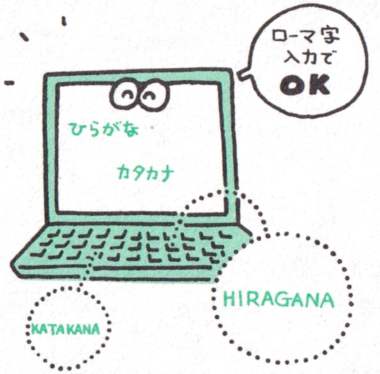
では、試してみよう。SHIFTキーを押しながらかなキーを押し、ランプが点灯したことを確かめてから、次のようにキーを押す。

ＡＳＡ→あさ
ＪＩＤＯＵＳＹＡ→じどうしゃ
ＧＡＫＫＯＵ→がっこう

次にカタカナを試してみる。CAPS LOCKキーを押して、以下のように打ってみよう。

ＰＡＳＯＫＯＮＮ→パソコン
ＫＯＮＮＰＹＵＸＴＡ→コンピュータ
ＧＵＲＡＦＩＫＫＵ→グラフィック

これでわかるように、「ＳＡ」で「さ」、「ＳＹＡ」で「しゃ」、促音（っ）は子音をダブらせるなどは、ふつうのローマ字表記と同じ。ただし、ローマ字とは書き方が違うものもあり、以下にまとめておく（例はひらがなで示したが、カタカナの場合も同じ）。

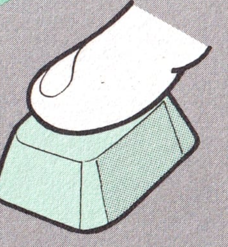


ん	N を2回続けて押す
小文字の母音（ぁぃぅぇぉ）	L とその母音を組み合わす（LA ～ LO）
ー（のばす音）	X を押す
ゔぁ ～ ゔぉ	V とその母音を組み合わす（VA ～ VO）
促音（っ）だけ	N、X 以外の子音キーを2回続けて押す

!?

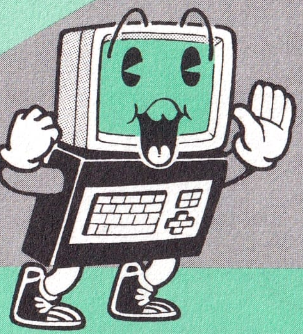
PART 3

プログラムをつくる前に知っておきたいこと



パソコンを使う場合、プログラムを自分でつくることが目的のひとつでしょう。PART 2まで学んだ基本を守りながら、ここからいよいよ、そのプログラムの作成にかかります。その前に、プログラム作成時の約束ごとをマスターしておきましょう。この章では、次の命令を学びます。

END(エンド)、**NEW**(ニュー)、**AUTO**(オート)、**RUN**(ラン)、**CONT**(コンティニュー)、**DELETE**(デリート)、**LIST**(リスト)です。



プログラムとはどんなものか

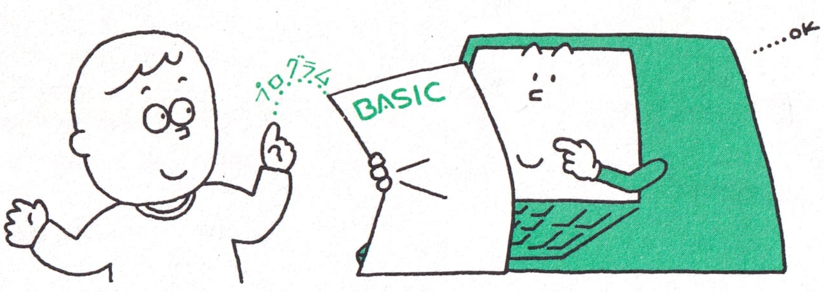
パソコンに仕事をさせる命令書

人に何かをしてもらうとき、私たちは「……してください」とことばをかけます。仕事が複雑なことだったり、口頭より確実に伝えたかったりする場合には、きちんと紙に書いて手渡します。学生に身近な、試験のことで考えてみましょう。各問ごとに、「計算をしなさい」、「簡単に答えなさい」という指示があります。この指示、あるいは命令をコンピュータにわかるように書いたものをプログラムと呼びます。いうまでもなく、プログラムはコンピュータ言語で書かれています。本書では、コンピュータ言語のうち、ベーシックで書いたプログラムのことを中心に解説します。

プログラムは、ふつう、ひとつの命令ではなく、いくつかの命令が集まったもので、これを与えることによって、パソコンはあるまとまった仕事をします。仕事の内容が複雑になるほど、長いプログラムになります。

ゲーム場などで、マイコンのゲームで遊んだことのある人も多いでしょう。あのゲームも、プログラムに従って動いているのです。約束ごとに従えば、プログラムをつくるのはだれにでもできます。ただ、コンピュータは、人間と違って、命令されたことしかしません。融通(ゆうずう)がきかないのです。ですから、プログラムは正確に書いてやる必要があります。

また、プログラムには、ふつうでは私たちがあまり使わない、独特のやり方が出てきます。これもコンピュータに正確に仕事をしてもらうためで、最初はなじみにくいかもしれませんが、本書を終わりまで読むころには、すっかり飲みこめるでしょう。



プログラムの基本形は「行番号＋命令文」

とにかく、プログラムの一例を見てみましょう。基本的には、すべて同じような形でできています。

```
10 C$="●"
20 X=29:Y=16
30 FOR I=0 TO 13
40     LOCATE X,Y:PRINT C$
50     LOCATE X,Y:PRINT " "
60     X=X-1:Y=Y-1
70 NEXT I
80 END
```

上の例でわかるように、1行ごとにまとまりを持っています。それぞれに命令が含まれているこのまとまりを命令文といい、各命令文の先頭についている番号を行番号といいます。この例では、行番号は10から始まって10刻みで80までついています。行番号のつけ方には、0～65529までの整数を使うこと以外に特別な決まりはないので、いくつから始めてもよく、刻みも自由です。

ただし、コンピュータは、原則として行番号の若い順に仕事をしていきます。この決まりを無視して行番号をつけるわけにはいきません。

例外もありますが、一般的には、プログラムの最後には終止命令である**END**（エンド）がきます。この命令は、プログラムの実行を終わらせる働きをします。

終止命令は、プログラムのどこに置いても、またいくつ書いてもかまいません。

```
行番号　命令文
行番号　命令文
行番号　命令文
  ⋮　　　⋮
行番号　END
（注）行番号は若い順になる。
```

▲プログラムの一般形式

基本形

END（エンド）

プログラムの実行を終わらせる終止命令



プログラム作成の手順は

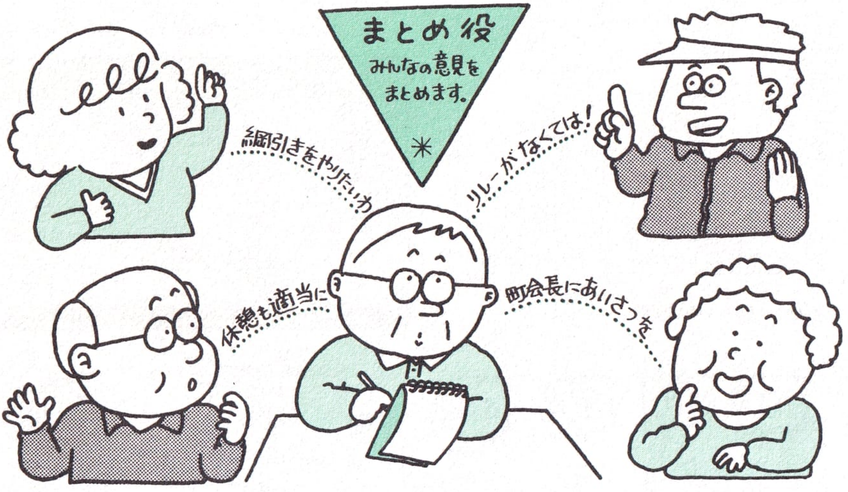
まず何をするかをまとめる

プログラムは複数の命令文の集まりであり、命令文は個々の命令から成り立っています。ただ、命令や命令文をたんに寄せ集めただけではプログラムとはいえません。同じ命令や命令文を使っても、順序を変えればまったく別の仕事をしてしまうのです。そこで、重要になるのは、命令文や命令の順序やつながりです。全体の仕事の流れといってもよいでしょう。これが、プログラムの作成の基本であり、また、むずかしいところでもあります。

仕事の流れをつかむには、まず「コンピュータにどんなことをさせるか」をはっきりと決めなくてはなりません。私たちも、ただ「買い物にいってきなさい」といわれても困ります。どこで、何を、どのくらい買ってくるのかを教えてもらわなければ、買い物にいけません。パソコンも同じです。仕事の内容をできるだけくわしく、明確に考えましょう。これを、課題設定と呼ぶことにします。

以下の図では、「町内運動会」をする場合になぞらえて、プログラム作成の手順を考えてみることにします。

1・課題設定



順序立てをするには図にするといい

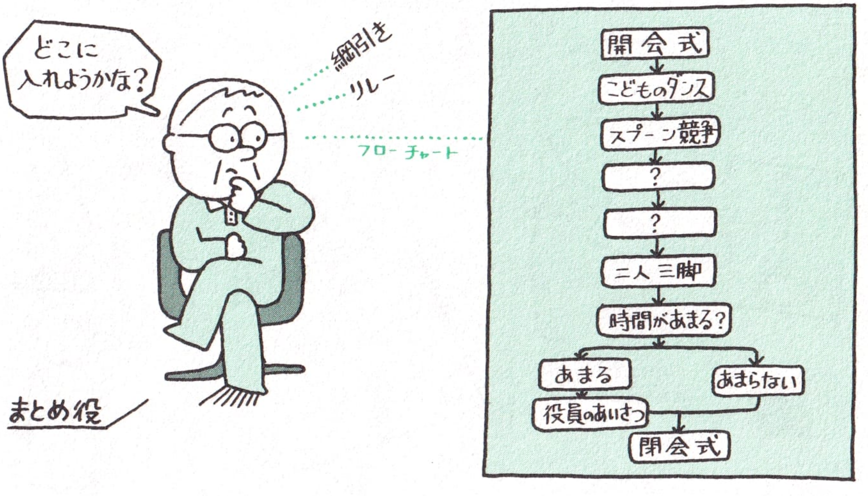
パソコンにどんな仕事をさせるかがはっきりしたら、こんどは「どんな命令を、どのような順序で実行すれば、目的とする仕事が達成できるか」を考えます。先の買い物の例でいえば、何を買い物してくるかを決めたあとで、どんな順序で店をまわればよいかを考えることになります。たとえば、次のとおりです。

①まず八百屋で玉ネギを1袋買う。
②魚屋でアジを3匹買う。
③パン屋で食パンを1斤(きん)買う。
④おつりがあったら、すきなお菓子を買ってもいい。

こんな単純なことならこれでもよいのですが、複雑なことなら、図にしたほうがずっとわかりやすくなります。プログラム作成の際には、一定の決まりに従って、図をつくることになります。この図をフローチャートといいます。記号や書き方が決まっているので、だれにでも理解できる利点があります(くわしくは46ページ参照)。

複雑な仕事の場合は、まず大まかなフローチャートを書き、それをしだいにくわしく、細かな内容にしていくという形をとります。こうすると、ポイントをおさえながら、ミスなくフローチャートができあがります。

2・フローチャート作成



命令をベーシックで翻訳する

フローチャートができあがったら、これをもとにプログラムを作成します。買い物だったら、買う順番にくわしいメモをつくってやることにあたります。お使いにいく人は、これに書かれているとおりに行動すれば、まちがいなく買い物をして帰ることができます。

フローチャートからプログラムをつくる作業を、コーディングといいます。ここでようやく、パソコンに対して、直接何かを行なう段階にきます。それ以前の課題設定やフローチャート作成は、ここまでの準備ですが、だからといって軽く考えてはいけません。この準備にいかに手間をかけ、いかに入念に準備をするかによって、プログラムのできあがりが速くも遅くもなります。また、プログラムの質そのものも、この準備の段階である程度決まってしまいます。

コーディングの作業は、ことばや図で「表示する」「計算する」などとあらわされている命令を、コンピュータにわかるようになおすことです。いわば、日本語をコンピュータ言語に翻訳する作業ともいえます。このときにポイントになるのは、ベーシックの命令を正しく理解して、ベーシックの決まり（文法）からはずれることなく使いこなすことです。それができれば、ここは思ったほどむずかしいことではありません。

3・コーディング



点検、ミスの訂正を忘れずに

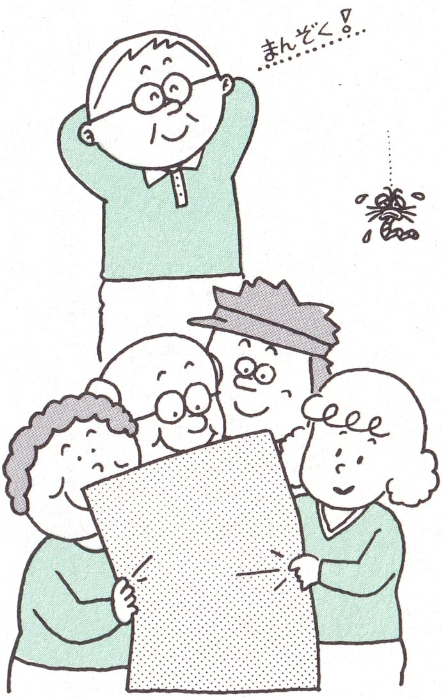
いちおうプログラムができあがったら、最初に考えたとおりの仕事をするかどうかを、実行して試してみます。この作業は必ず行ないましょう。単純なミスならパソコンが「まちがっているよ」と知らせてくれますが、考え方、命令の組み立て、手順などのまちがいは、パソコンでもわかりません。「できた！」と喜ぶ前に、ミスを探してください。長いプログラムはもちろん、短いプログラムにもミスは起こります。プログラムの長短にかかわりなく、点検を習慣づけます。

プログラムのミスを、バグといいます。バグとは虫のことで、プログラムに小さな虫がいるために、おかしなことになる、というわけです。ミスを修正することを、デバッグといいます。バグが虫ですから、デバッグは虫とり、ということです。これでバグなしと判断すれば、プログラムは完成です。場合によっては、何度も使っているうちにバグを発見し、デバッグが必要になることもあります。

4・デバッグ



5・完成

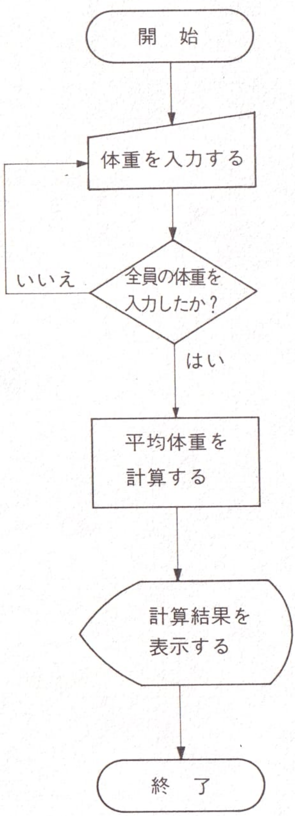


フローチャートの書き方

フローチャートを書くには、フローチャート記号といって、JIS 規格で定められている専用の形を使います。フローチャート記号は全部で 30 種類ありますが、よく使われるのは下表のとおりです。

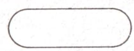
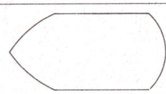
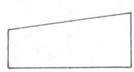
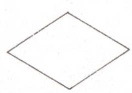
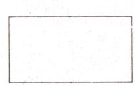
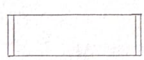
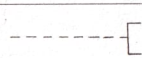
一例として、「何人かの平均体重を計算して、結果を表示する」という処理のフローチャートを示しておきます。ただしこれは、あくまでも概略で、実際は、もっと複雑なもの（すぐにプログラム化できるもの）にする必要があります。

本書でもフローチャートを何度も使いますが、わかりやすくするために、開始の記号を省略したり、処理内容をまとめたりして、通常書かれるより簡略にしてあります。



▲フローチャートの例

◉よく使われるフローチャート記号

	プログラムの開始、終了、あるいは中断をあらわす
	画面に表示することをあらわす。表示される内容は、見出し、計算結果、選択肢など
	キーボード操作をするとき、使う記号。具体的には、キーボードからデータを入力するときなどに使う
	処理の流れが、条件によって変わるときに使う。ふつう、条件の内容が、それに対する判断とともに示される
	一般的な処理に用い、処理の内容をあらわす。処理にはデータの演算、記憶、移動、編集などがある
	別の場所で定義、処理した命令（サブルーチンなど）をあらわすときに使う
	処理の内容を説明したり、注意を与えるときに使う
（矢印）	処理の流れをあらわす。プログラムは、この矢印が示す順序で実行される

プログラムを入力する

ミスのない入力の4大原則

プログラムを入力するとき、必ず守ってほしい原則が4つあります。この原則を破ると、入力しようとするプログラムがうまく入力できないばかりか、すでに入力してあるプログラムをこわしてしまうこともあります。

✣原則①＝まず画面を消すこと（SHIFT ＋ CLS/HOME）

これから入力しようとするプログラム以外の文字や記号があると、まちがいの原因になります。すべて消してから、新しい入力にとりかかります。それには、SHIFT を押しながら画面消去キー CLS/HOME を押します。

✣原則②＝記憶装置 RAM の内容をすべて消す（記憶消去指令 NEW＋ RETURN ）

入力したプログラムは、RAM に記憶されます。そこに古いプログラムがあると、新しいプログラムとまじりあってしまいます。そこで古い記憶を消すためには、記憶消去指令である **NEW** をキーインし、そのあと必ず入力終了キー RETURN を押します。

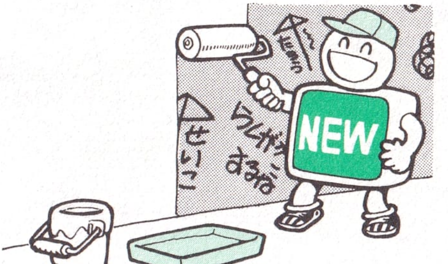
✣原則③＝1行は255文字以内にする

プログラムの1行とは、行番号を打ちこみはじめてから入力終了キー RETURN を押すまでをいいます。画面での行数とは関係ありません。画面を消し、RAM の内容を消去したら、いよいよプログラムを入力していきますが、MSX ベーシックで1行に入力できる文字数は、行番号や空白なども含めて255文字以内です。

✣原則④＝1行ぶんキーインしたら、必ず入力終了キー RETURN

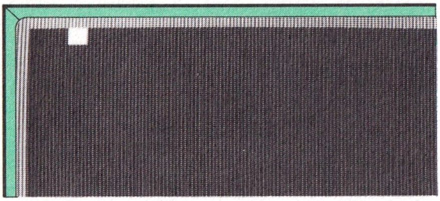
1行ぶんキーインしたら、必ず入力終了キー RETURN を押し、カーソルが次の行の先頭にきたのを確認し、続けて入力していきます。

基本形
NEW
RAM の内容を消す記憶消去指令



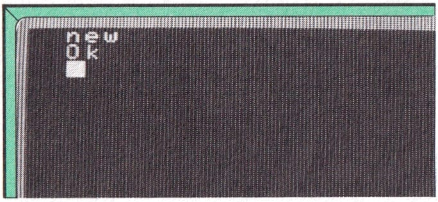
例 1行ぶんを入力する

① SHIFT を押しながら CLS HOME キーを押す



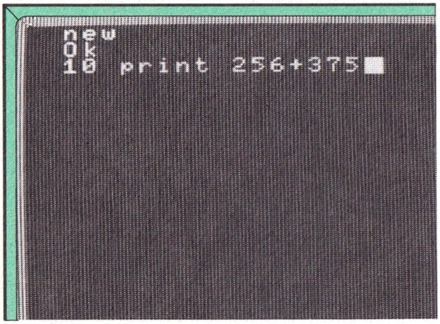
画面の文字が消え、カーソルは画面の左上端に移動します。

② N E W とキーインして、RETURN キーを押す

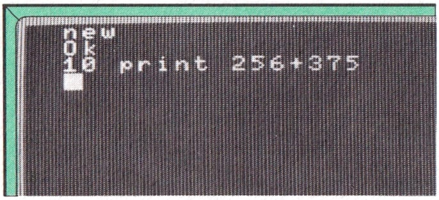


「Ok」が出たら記憶消去は完了です。

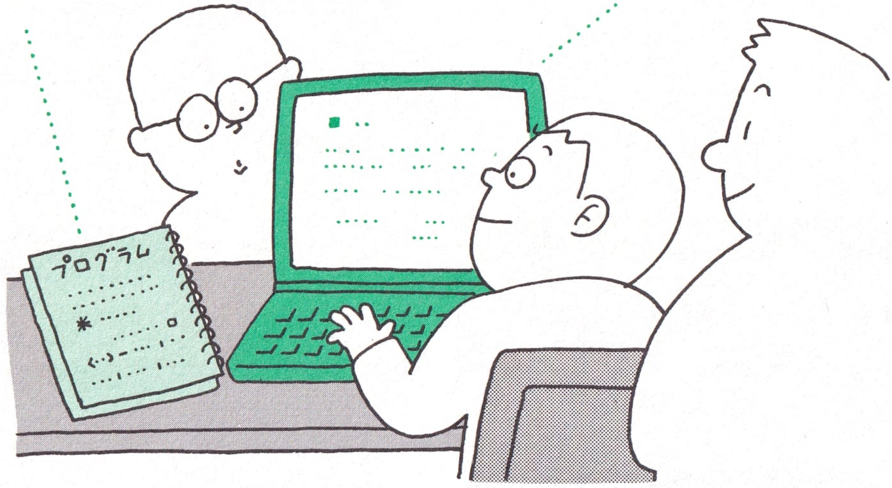
③プログラムをキーインする



④ RETURN キーを押す



カーソルが次の行の左端に移動。これで1行ぶんの入力は終わりです。続いて、次の行を入力します。



能率よく入力するための5つのポイント

プログラムの入力は、めんどうくさい作業です。とくに初心者にとっては、キーの配列に慣れていないので、目的とする文字を探しだすだけでもひと苦労でしょう。ポツリ、ポツリとしか打てず、イライラするかもしれません。キー操作は、慣れれば自然に速くなりますが、それにしても、できるだけ手間のかからない方法をとるほうがいいに決まっています。そのためのポイントをあげます。

✣ポイント①＝命令語のあとは、1字ぶん以上の余白をとる

余白をとるのは、見やすくすることが目的です。そうしないと入力できないとか、エラー表示が出るということはありませんが、余白をとる習慣をつけておいたほうがよいでしょう。

本書では、余白を1字ぶんとし、「␣」であらわしています。余白をとるには、上に何も表記されていない空白キーを使います。

✣ポイント②＝行番号は10刻みで

行番号は、0～65529までの整数なら、自由に選んでつけることができます。ただし、一般には10から始めて10刻みで、10、20、30、40……とつけるほうが便利です。

その理由は、まず見やすいこと。それに、あとからプログラムを追加して割りこませることができるからです。たとえば行番号20と30のあいだにあと数行追加する必要が生じても、21～29の9行ぶんだけ余裕があります(58ページ参照)。

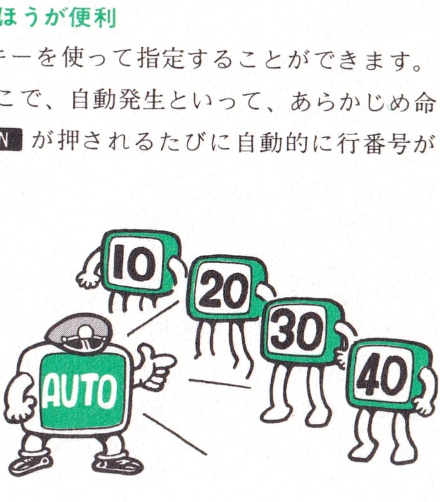
✣ポイント③＝行番号は自動発生させたほうが便利

行番号は、[1][0]というように、数字キーを使って指定することができます。しかし、それでは手間がかかります。そこで、自動発生といって、あらかじめ命令を入力しておき、入力終了キー [RETURN] が押されるたびに自動的に行番号があらわれる方法をとります。

基本形

AUTO(オート)␣行番号，増しぶん（刻み）

行番号を自動的に表示する行番号指令。
AUTO(オート)␣のかわりに、機能キー [F2] を使うこともできる



それには、行番号指令（AUTO）を使い、最初の行番号と増しぶん（刻み）を指定します。

基本形の中で、行番号を省略すると自動的に0、増しぶんを省略すると自動的に10刻みとなります。また、行番号、増しぶんとも省略すると、行番号は10から始まり、10刻みで自動発生します。つまり、AUTOを使って行番号を指定するには、次の4種類の方法があることになります。

①AUTO␣100，5――行番号は100、105、110、115……

②AUTO␣100――行番号は100、110、120、130、140……

③AUTO␣，5――行番号は0、5、10、15、20……

④AUTO――行番号は10、20、30、40……

この方法でも、1行ぶんの入力を終えたあとは必ず RETURN キーを押し、入力が終了したことを告げます。なお、行番号の自動発生を解除するには、組み合わせキー CTRL を押しながら、停止キー STOP （または C ）を押します。

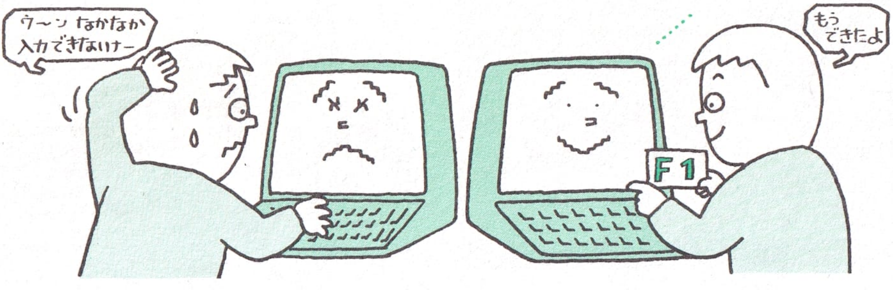
ポイント④＝プログラムの文字は小文字にする

プログラムは、大文字でも小文字でもかまいません。コンピュータはどちらでも同じように理解します。それなら、小文字のほうが、 CAPS LOCK キーを押さなくてすむだけ便利です。本書では、プログラムはすべて大文字になっていますが、君たちが入力するときは小文字でよいでしょう。

ポイント⑤＝機能キーをどんどん使う

機能キー F1 ～ F10 には文字が割りあてられているので、4、5回のキー操作が1回ですみます。大いに使いましょう。ポイント③で述べた行番号指令AUTOも、 F2 にセットされています。最初が10で、10刻みの行番号を出すのなら、このキーをひとつ押すだけです。

ほかにも、プログラム入力時に使える文字がセットされているので、一覧表にして、いつでもわかるようにしておくとよいでしょう(32ページ参照)。



プログラムの実行と停止

プログラムの実行は実行指令 RUN で

プログラムを入力したら、それをもとに、プログラムどおりの仕事をさせます。これを実行といいます。実行には、実行指令 **RUN**(ラン) を使います。この指令を実行することにより、RAM(ラム) に記憶させたプログラムを実行することができます。

RUN(ラン) の次に行番号を指定すると、その行番号から実行を始めます。行番号の指定がなければ、もっとも若い行番号から実行することになります。この指令を実行するには、基本形をキーインしたあとで、入力終了キー RETURN を押します。

また、実行指令はよく使われるので、F10/F5 キーにセットされています。このキーは入力終了キーの働きも含んでいるので、F10/F5 キーを押したあとで、RETURN キーを押す必要はありません。

なお、画面には入力したプログラムが表示されていますから、実行の前にすべて消去しておくことにします。以上をまとめると、プログラム実行は次のように行ないます。

基本形

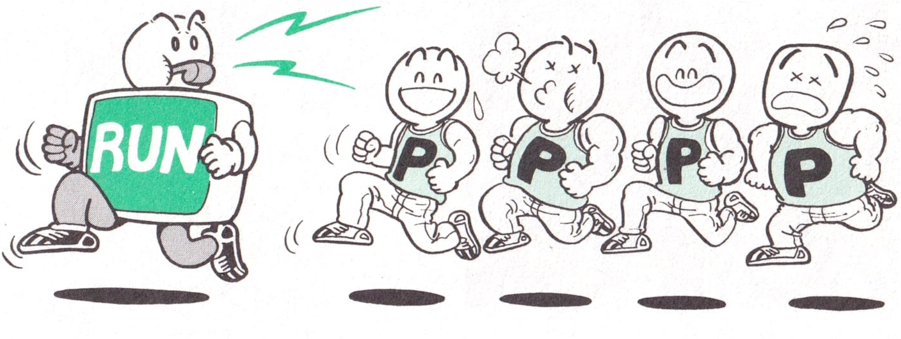
RUN(ラン)␣**行番号**

入力したプログラムを実行させる実行指令。RUN(ラン) のかわりに F5 を使うこともできる

① SHIFT キー＋ CLS/HOME キーを押す。

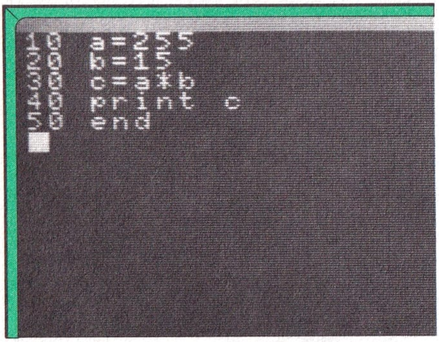
② R U N とキーインして RETURN キーを押す。または、F10/F5 キーを押す。

SHIFT キーを押しながら F10/F5 を押すと（F10)、①と②をひとつのキーですませることができる。

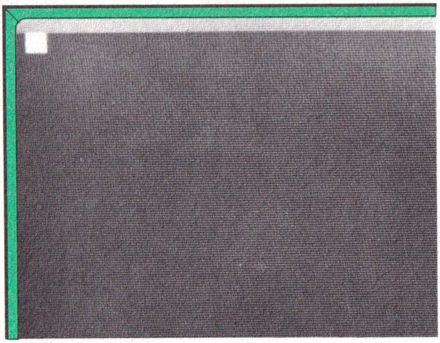


例 販売個数（a）と販売単価（b）から販売額（c）を計算するプログラムを実行する

①プログラムを入力する



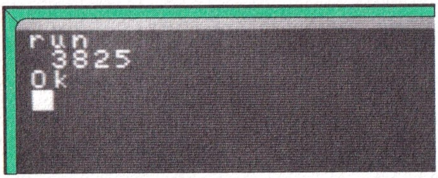
② SHIFT を押しながら CLS HOME キーを押す



③ R U N をキーインする



④ RETURN キーを押す



プログラムが実行され、実行が終わると「Ok」が表示されます。

F5 キーを使った場合には、③以降は次のようになります。

⑤ F5 をキーインする



プログラムにミスがあれば、画面にはエラー表示と「Ok」が出て、その下にカーソルが点滅し、実行が終了します。

プログラムの実行を停止するには

何かの理由でプログラムを停止したいときには、2つの方法があります。一方は一時的に停止する場合で、他方は停止すると同時に、強制的に終了させてしまう中断の場合です。

✣一時停止は停止キー STOP を使う

プログラムを実行している途中で、一時的に実行を止めたい場合には、停止キー STOP を押します。すぐに停止します。画面上の変化は、プログラムの内容や停止キーを押すタイミングによって違います。

実行を再開するときには、もう一度停止キー STOP を押します。一時停止中には、停止キー以外のキーを押しても入力されません。

✣中断させてしまうには CTRL を押しながら STOP

プログラムの実行を強制的に終わらせてしまうには、組み合わせキー CTRL を押しながら、停止キー STOP を押します。これで実行は終了し、画面には「Break in ××」（××は数字）が出て、「Ok」表示とカーソルが出ます。「Break in ××」は「行××で実行を停止した」という意味です。

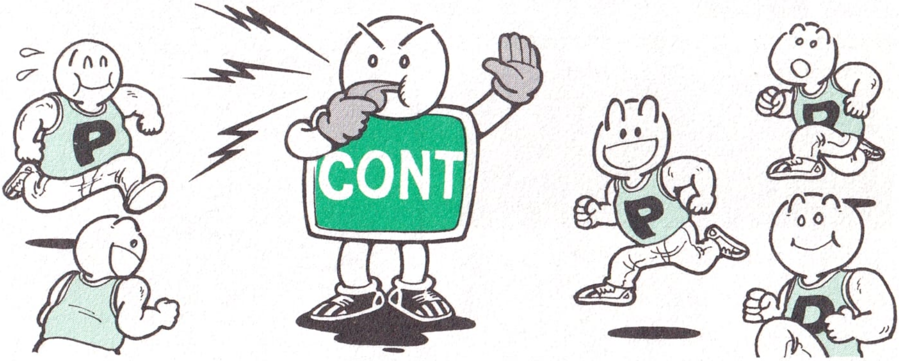
基本形

コンティニュー
CONT

中断したプログラムの実行を再開する再開指令。機能キー F8 にセットされている

再開するには、再開指令 CONT（コンティニュー）を使います。C O N T RETURN で、プログラムは再び中断した次の行から実行を始めます。

機能キーでは F8 にセットされています。SHIFT を押しながら F8/F3 を押せば、同じ働きをします。



電卓と同じような使い方も可能

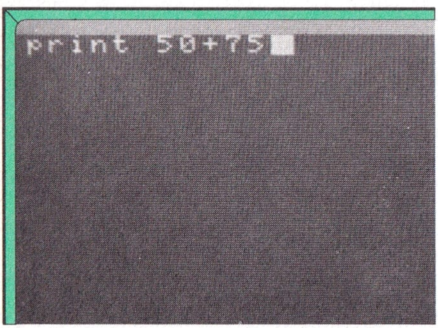
直接モードによる実行

これまで述べてきたように、入力したプログラムはいったん記憶され、実行指令 **RUN**（ラン）によって実行されます。このような仕事の方法を「間接モード（プログラムモード）による実行」といいます。これが、コンピュータ本来の仕事の方法ですが、ほかに「直接モード（ダイレクトモード）による実行」という方法もあります。

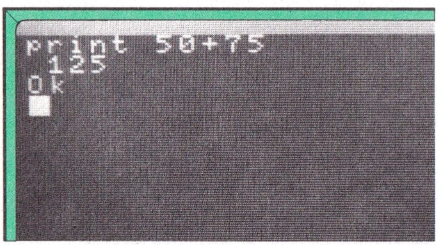
これは、コンピュータに与えた命令を、記憶させずにすぐに実行する、というやり方です。つまり、電卓と同じような使い方だといえます。

例　「50＋75」を直接モードで実行し、結果を表示する

①PRINT␣50＋75 とキーインする

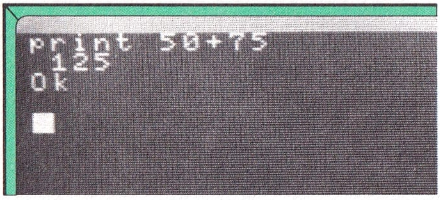


② RETURN キーを押す

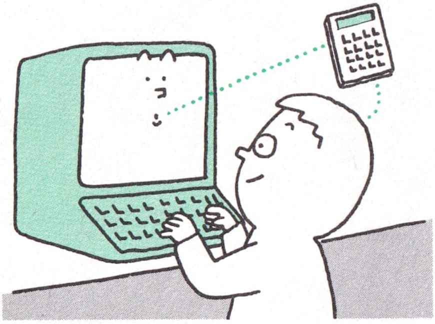


実行指令を入力しなくても、きちんと計算されて答えが出ます。

③再び RETURN キーを押す



計算されず、カーソルが1行ぶんさがるだけです。つまり、直接モード実行は、1回だけです。



間接モードと直接モードの違いは

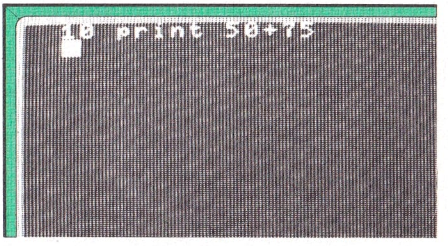
間接モードと、直接モード実行の表面的な違いは、行番号の有無です。直接モード実行の命令文には、行番号がありません。これに行番号をつけると、間接モード実行になるわけです。

例　「50＋75」を間接モードで実行し、結果を表示する

①10␣PRINT␣50＋75 とキーイン

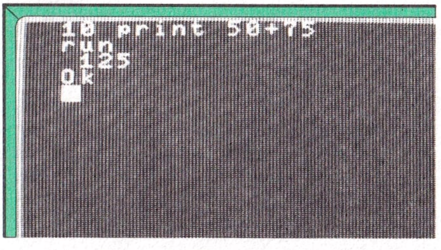


② RETURN キーを押す



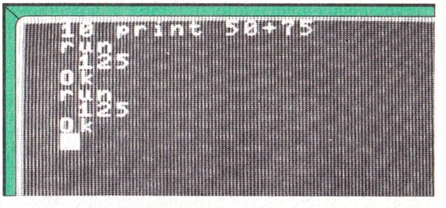
まだプログラムは実行されません。

③ F5 キーを押す



プログラムが実行されました。

④再び F5 を押す



何度くり返しても、同じようにプログラムは実行されます。

◉直接モードと間接モードの相違

相違点	直接モード実行	間接モード実行
行　番　号	な　し	あ　り
記　憶	記憶されない	記憶される
実行の方法	入力終了キーで実行	実行指令で実行
実行のくり返し	不　可　能	何回でも可能

もしエラーメッセージが出たら

誤りのある行数が表示される

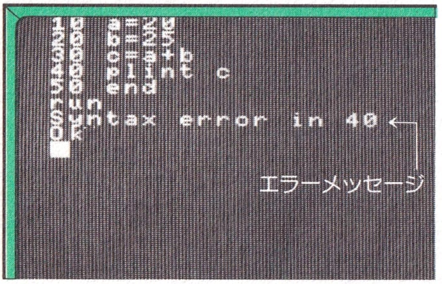
MSX パソコンは、プログラムや命令文のまちがいを、1 行ずつ実行しながらチェックする機能を持っています。誤りのあるプログラムや命令文を実行するとブザーが鳴り、画面にエラーメッセージが出て、実行が終了します。間接モードでは、誤りのある行の番号も表示されますから、誤りのある箇所はすぐに発見できます。

「Syntax error」は、文法上の誤りがあることを意味するエラーメッセージで、in のあとの数字が誤りのある行番号です。下の例では、よく見ると、本来「print」であるべき命令が「plint」になっています。

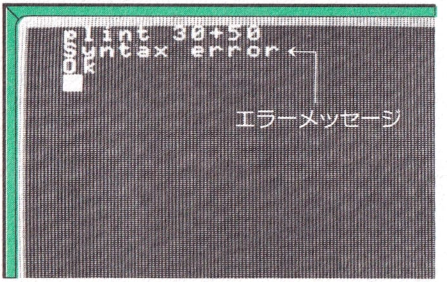
エラーメッセージは、内容により 35 種類あります（207 ページ参照）。下の例では誤りのある行の行番号が表示されていますが、種類によっては表示されません。

間接モードでは実行指令を入力し、実行してからエラーメッセージが出ますが、直接モードでは、入力終了キーを押すとすぐにエラーメッセージが出ます。

例



例



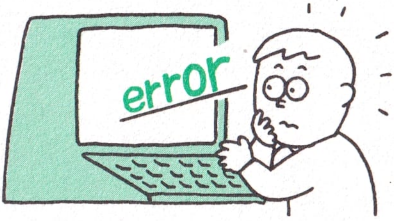


エラーを訂正する手順

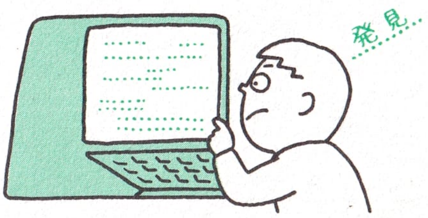
間接モード実行でエラーメッセージが出たら、誤りを見つけて修正します。手順のうち、❹は忘れやすいので要注意。これを忘れて再実行すると、再び同じエラーメッセージが出ます。入力終了キー RETURN を押す際、カーソルは、その行の中にあればどこにあってもかまいません。入力終了キーを押せば、カーソルのある行全体が入力されます。

なお、訂正がうまくいかず、何度もエラーメッセージが出る場合は、画面消去キー CLS HOME と記憶消去指令 **NEW** で画面と記憶を白紙の状態にして、もう一度初めからやりなおすとよいでしょう。

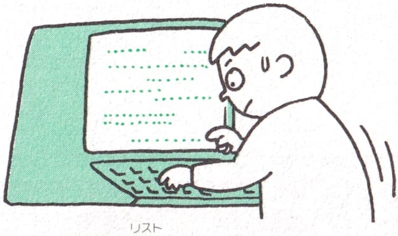
❶エラーメッセージをよく見て、誤りの場所と内容を確認する。



❷誤りのある行を点検し、誤りを見つける。



❸文字の訂正、挿入、削除、追加の方法で誤りを訂正する。



❹誤りを訂正したら、入力終了キー RETURN を押す。



❺リスト指令ＬＩＳＴでリストをとって確認する。



❻実行指令ＲＵＮによって再実行する。



▲エラー表示の処理手順

プログラムの追加、削除の方法

追加は CLS HOME と新しい行＋ RETURN で

プログラムの追加手順は、次のとおりです。

① SHIFT を押しながら画面消去キー CLS HOME を押して、画面をすべて消す。

②追加する位置の前の行と後ろの行とのあいだの行番号を新しくつけて、追加するプログラムをキーインする。

③ RETURN キーを押す。

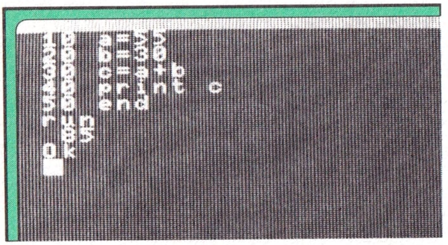
これで追加する行は入力され、行番号の若い順に整理されて記憶されます。これは画面ではわかりませんが、実行すればわかります。

同じようにして、行番号に注意すれば、何行でも追加できます。追加する行の行番号は、前後の行のあいだの数ならどんな数でもかまいません。前出の例では35としましたが、31や33でも、あるいは38や39でも同じ結果になります。

なお、すでに入力してあるプログラムの中の行と、同じ行番号をつけた新しい行を入力すると、古いほうの行の内容は自動的に消え、新しい行の内容に変わります。この機能は、1行全体を修正するのに利用すると便利です。

例 A、Bの値を表示するプログラムを追加する

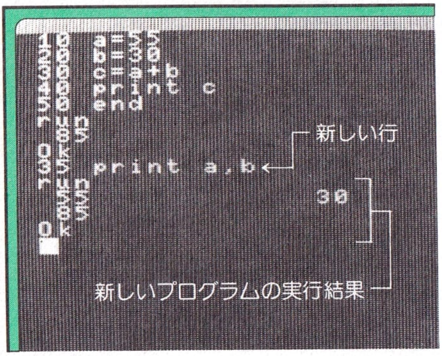
①もとのプログラムと実行結果



このプログラムの行30と行40のあいだに、「print␣a, b」を追加します。

②追加する行の行番号を35として、キーインし、 RETURN キーを押す

③ F5 キーを押して実行する



※画面消去をせずに追加

行ごとの削除には3つの方法がある

プログラムの一部を削除する方法は、組み合わせキー CTRL を使う方法、入力終了キー RETURN を使う方法、削除指令 DELETE(デリート) を使う方法の3とおりあります。

削除の方法❶＝ CTRL を使う

①削除する行の行番号のすぐ後ろにカーソルを移動する（たとえば行 40 を削除する場合は「40■」)。

② CTRL キーを押しながら E キーを押す。

③ RETURN キーを押す。

削除の方法❷＝ RETURN キーを使う

①削除する行番号をキーインする。

② RETURN キーを押す。

以上の方法（❶、❷）は、削除する行が少ないときに用います。とくに前者の方法（ CTRL キーを使う）は、画面上で、その行の内容を見ながら削除を行なうことができるため、ミスも少なくて便利です。

削除の方法❸＝削除指令 DELETE(デリート) を使う

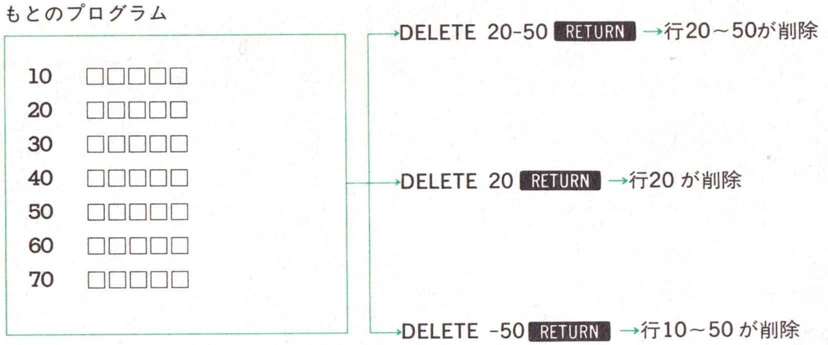
削除指令による削除は、削除する行がとびとびではなく、連続している場合に使います。削除指令は DELETE(デリート) です。

基本形のうち、―（ハイフン）でつないだ2つの行番号は、左側が削除する部分の初めの行の行番号、右側が最後の行の行番号です。

ハイフンと右側の行番号を省略して、「DELETE(デリート)␣行番号」とすると、その行番号だけが削除されます。また、左側の行番号を省略して「DELETE(デリート)␣―行番号」とすると、そのプログラムの最初の行から指定した行番号のあいだにある行が、すべて削除されます。

基本形
DELETE(デリート)␣行番号―行番号
プログラムの指定した部分を削除する削除指令

削除指令を実行するには、削除指令をキーインしたあと、入力終了キー RETURN を押します。入力終了キーと削除指令による削除では、削除した行は画面に残っていますが、コンピュータの記憶装置（RAM(ラム)）からは削除されているので、実行指令を行なっても実行されません。



▲削除指令 DELETE の使い方と実行結果

C/O/L/U/M/N

◉コンピュータは白か黒かの世界◉

コンピュータはたくさんのスイッチからできている。スイッチだから、判断できるのは電気が通っているかどうかだけだ。たとえば、電気回路に電流が流れていれば黒、流れていなければ白としておく。こんなしくみをいくつも並べると、白と黒の組み合わせだけでいろいろ複雑なことをあらわすことができる。

この、白と黒とを組み合わせたスイッチを1ビットという。電気製品などのスイッチと同じようなものだと思えばいい、白を**ON**、黒を**OFF**と考えれば同じことだ。

この1ビットが、コンピュータで扱える最小限の大きさ。これで、どのくらいの表現ができるか考えてみよう。

たとえば、わかりやすい例として数字のことで説明する。

われわれがふだん使っているのは、10進法だ。ここで、その0～5を、白と黒のスイッチであらわしてみる。次に、白を0、黒を1におきかえれば、10進法ではない、2進法の数字になる。このように、すべての数は2進法であらわすことができる（別表）。

10進法と2進法を比べると、人間には10進法のほうが手間がかからない。が、2進法のほうが単純な形で処理ができるので、コンピュータには適していることになる。

0→○○○○○○○○
1→○○○○○○○●
2→○○○○○○●○
3→○○○○○○●●
4→○○○○○●○○
5→○○○○○●○●

⇩ 2進法に変換

0＝00000000
1＝00000001
2＝00000010
3＝00000011
4＝00000100
5＝00000101

リスト指令でプログラムを確認

リスト指令 LIST には5つの使い方がある

プログラムに訂正、追加、削除などの操作で手を加えても、実際に画面で表示されているものと、RAM に記憶されている内容は一致しません。画面では訂正や追加が正しく行なわれたかどうか、確認できないのです。これを確認するためには、RAM に記憶されている内容をすべて画面に表示する、リスト指令 **LIST** を使います。

また、特別な操作を加えない場合でも、入力したプログラムが実際に記憶されたかどうかを確認するために、リスト指令を使うこともあります。

「—」(ハイフン) でつないだ 2 つの行番号は、左側の行番号から右側の行番号までのプログラム表示を指定するためのものです。指定のしかたにより、表示される部分が次のように変わります。

このうち、よく使うのは **LIST** のみですべてを表示する方法です。エラーメッセージが出たときには、まず誤りのある行だけを表示し、誤りを検出、訂正することもあります。

基本形

LIST␣行番号—行番号

RAM に記憶させたプログラムを画面に表示する。**LIST** のかわりに、[F4] を使うこともできる

記憶されているプログラム

```
10  □□□□□
20  □□□□□
30  □□□□□
40  □□□□□
50  □□□□□
60  □□□□□
 ⋮
```

- →LIST␣30-60 [RETURN] →行 30~60 が表示
- →LIST␣-60 [RETURN] →行 10~60 が表示
- →LIST␣30- [RETURN] →行 30 から最終行までが表示
- →LIST␣30 [RETURN] →行 30 が表示
- →LIST [RETURN] →記憶されているプログラムがすべて表示

▲リスト指令 LIST の使い方と実行結果

LIST(リスト)でプログラムをすべて表示する手順

① SHIFT を押しながら CLS/HOME を押して画面を消去する。

② L I S T とキーインする。

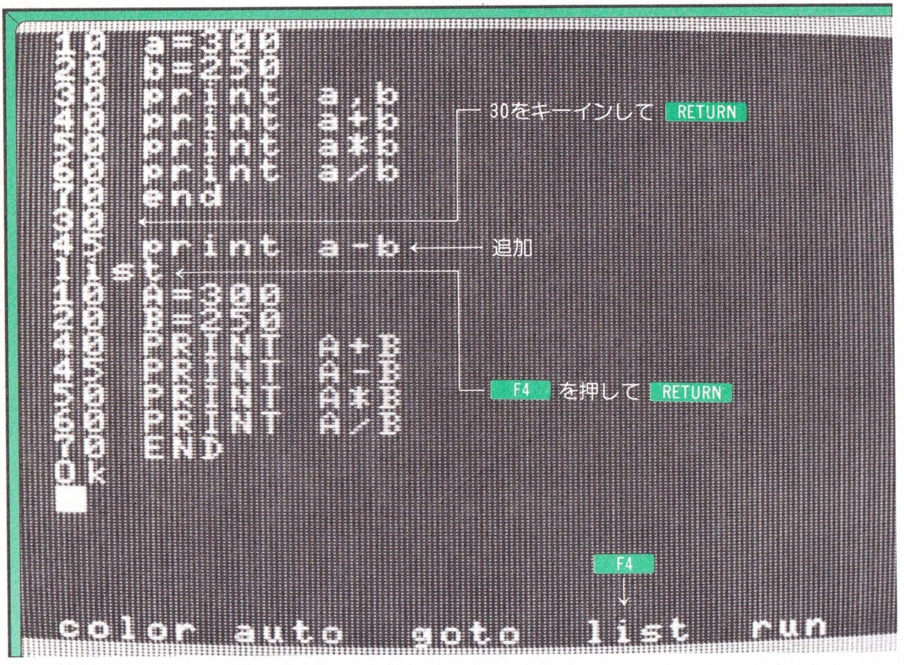
③ RETURN キーを押す。

これでプログラムが表示されます。リスト指令は入力終了キー RETURN を押さないと実行できないことを頭に入れておいてください。リスト指令は F9/F4 キーにセットされていますから、LIST とキーインするかわりに F9/F4 キーを押しても同じことです。こうすると、手順が簡単になります。

リスト指令で表示されるプログラムは、追加した行も含めて、すべて行番号の若い順に並んでいます。また、とり出したリストの文字は、すべて大文字になります。

なお、リスト指令により画面に表示されたプログラムをプログラム・リストといい、このような操作を「リストをとる」といいます。

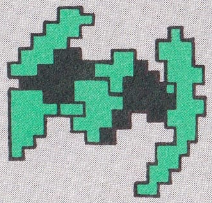
例 プログラムを削除、追加したあと、プログラム・リストを表示させる

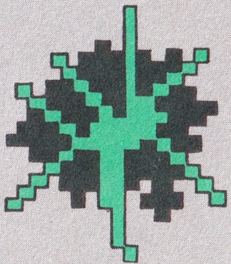


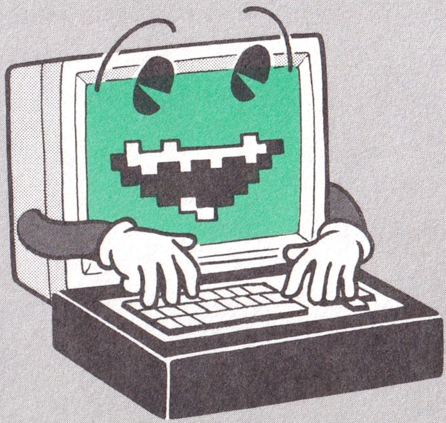
・PART・ 4

ゲームづくりの基本

記号を動かす







この章では、グラフィック・シンボルを思いどおりに動かすプログラムをつくります。いわば、コンピュータ・グラフィックスの基礎の基礎です。また、ゲームなどでよく使うジョイスティックも、簡単な方法で試してみましょう。使用する命令も少なく、プログラムもそれほど複雑なものではありません。座標の考え方をきちんと把握することがポイントです。

この章で学ぶ命令は、LOCATE（ロケート）、PRINT（プリント）、FOR〜NEXT（フォー・ネクスト）、GOTO（ゴーツー）、RND（ランダム）、INT（インテジャー）、STICK（スティック）です。

LESSON—①

画面上の位置を指定する

画面はグラフ用紙のようなもの

この章の最終目的は、「記号を自由自在に動かす」ことです。そのための手順を考えてみましょう。

①記号を画面に表示する位置を決める。

②記号を画面に表示する。

③表示された記号の位置を変化させる。

大まかにいえば、この3つの手順が必要です。まず①の位置指定の方法から説明しましょう。

MSX 2パソコンは、大きく分けて、2種類の画面を持っています。1枚は、キーボードから入力した文字やプログラム・リストなど、文字だけしか表示できないテキスト画面、もう1枚は点や線などでできた絵も表示できるグラフィック画面です。テキスト画面には絵は表示できませんが、グラフィック画面に文字を表示することはできます。

さらに、スプライト機能を使えば、このほかに32枚のスプライト画面を使えます。どの画面も、縦と横の線で、マス目のように区切られています。マス目の数は画面の種類によって違いますが、どの画面もちょうどグラフ用紙のような形になっていると考えればよいでしょう。文字や点は、このマス目ごとに表示されることになっています。

ここで説明するのは、文字を表示するテキスト画面です（グラフィック画面とスプライト画面については後述）。テキスト画面は、マス目の数により、さらに2つに分かれます。

種　　類	表示できる最大の文字数	ふつう表示できる文字数
テキスト画面 I	横40字×縦24行 (横80字×縦24行)	横39字×縦24行
テキスト画面 II	横32字×縦24行	横29字×縦24行

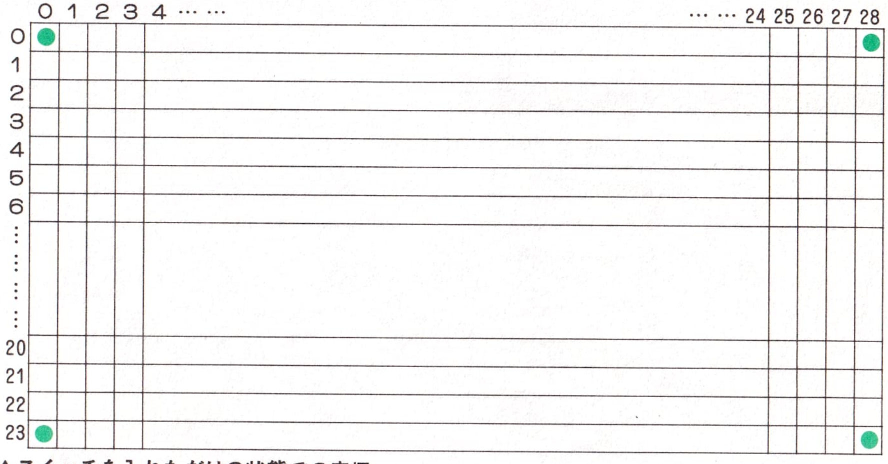
表示位置の指定は LOCATE（ロケート）命令で

MSX 2 パソコンは、電源スイッチを入れた直後はテキスト画面IIのふつうの状態、つまり横29字ぶん、縦24行ぶんの文字が表示できる状態になっています。画面いっぱいに文字を表示すると、29×24で 696 文字になります。1文字ぶんがひとつのマス目で、全体がグラフ用紙のようになっているのですから、1つひとつのマス目の位置は、下図のように座標としてあらわせます。たとえば、左上のすみは（0, 0）、右上のすみは（28, 0）、左下のすみは（0, 23）、右下のすみは（28, 23）になります。0から始まっているので、横は0～28、縦は0～23の範囲であることに注意してください。

画面上の位置を指定するには、位置指定命令である **LOCATE**（ロケート）を使います。たとえば、左下のすみに表示させたいのなら「LOCATE␣0, 23」となります。ただし、この命令には表示する働きがありません。位置を指定するだけです。

> **基本形**
>
> **LOCATE（ロケート）␣水平座標，垂直座標**
>
> 入力した文字をどの位置に表示するかを指定する位置指定命令

画面の範囲外に座標を指定すると（たとえば LOCATE␣100, 500）、「Illegal（イリーガル） function（ファンクション） call（コール）」というエラーメッセージが出ます。



▲スイッチを入れただけの状態での座標

画面に表示するのは PRINT 命令

次に手順の２番め、記号を画面に表示させる方法について説明しましょう。キーボードから入力すれば、文字は画面に表示されますが、これではプログラムにはなりません。直接モードと間接モードのところで説明したように、画面に表示されたからといって、パソコンの記憶装置に入っているわけではないので、くり返して実行することは不可能です。

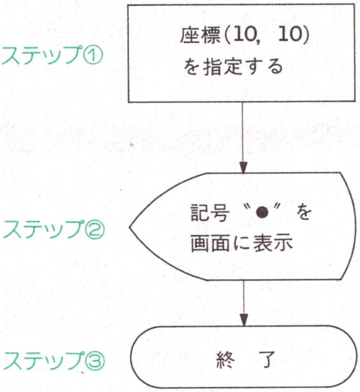
文字や記号を画面に表示させるプログラムをつくるには、PRINT という表示命令を使います。この命令文を実行すると、２つの「" "」（ダブルコーテーション）で囲まれた文字や記号がそのまま表示されます。後ろのダブルコーテーションマークは、つけなくてもかまいませんが、本書ではすべてきちんとつけることにします。

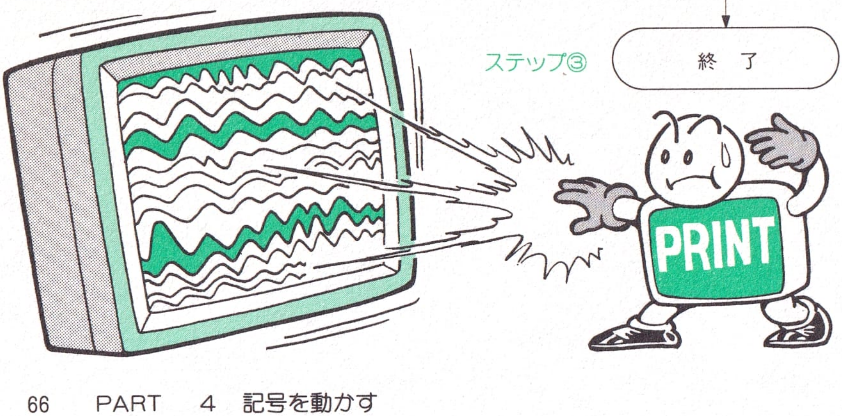
さて、これで先の手順の①と②の課題を行なえる命令がそろいました。ここでは、「座標（10, 10）の位置にグラフィック・シンボルの●を表示する」プログラムをつくることにします。手順は簡単ですが、練習のためにフローチャートを書くことにしましょう。

基本形

PRINT␣"文字または記号"

文字や記号を画面に表示する表示命令





座標(10, 10)に●を表示するプログラム

前項に示したフローチャートをもとに、プログラムをつくってみましょう。各ステップで使う命令は、次のとおりです。

①座標（10, 10）の指定―― LOCATE␣10, 10

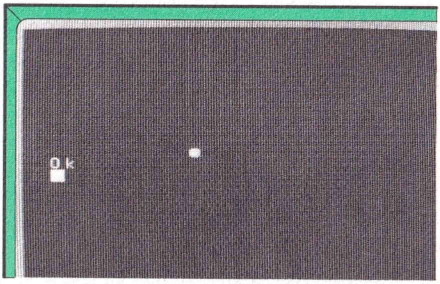
②記号●の表示―― PRINT␣"●"

③プログラムの終了―― END

このあとは、それぞれの前に行番号をつければできあがりです。

```
10 LOCATE 10,10
20 PRINT "●"
30 END
```

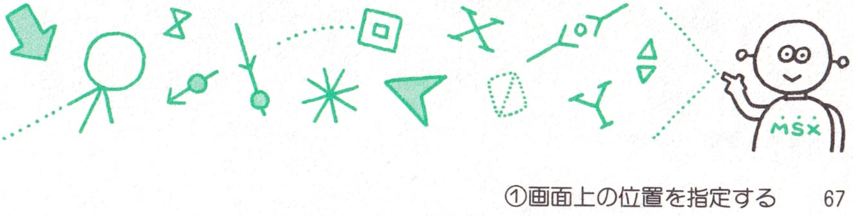
グラフィック・シンボルをどのキーで入力するか、わからない場合は31ページの図を見てください。「●」は、[ケ]のキーで入力できます。コンマ（，）、ダブルコーテーション（" "）、命令のつづりなどに抜けやまちがいがないことを確認したら、SHIFTを押しながらF10/F5を押します（F10になる）。プログラムが実行されて、下の画面のようになるはずです。



この位置が座標（10, 10）で、左から11字め、上から11行めになっています（座標の番号は0から始まるので、座標の値は字数、行数より1つ少ない）。座標の指定やグラフィック・シンボルを変えれば、望む位置に違う記号を表示することができます。

練習問題 3

グラフ用紙に座標を書き、それを見ながら**LOCATE**（ロケート）で指定した位置に、自分のすきな数字や文字、グラフィック・シンボルを入力してみよう。



LESSON ❷

画面上の記号を移動させる

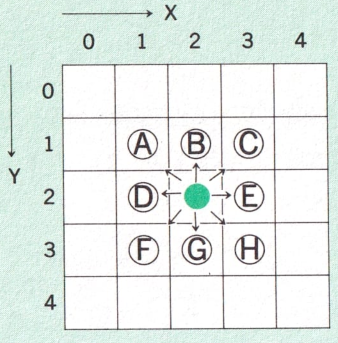
座標を変えれば記号が動く

では、いよいよ画面に表示させた記号を動かすプログラムにとりくみましょう。ここでもまた、座標の考え方が基本になります。記号を表示する位置の座標を変化させれば、それに従って記号は移動します。

図のように、水平座標Xの値と、垂直座標Yの値を操作することによって、横、縦、斜めのどの方向にも、自由に移動させることができます。横に動かすにはYを固定しておいてXを変化させ、縦に動かすにはXを固定しておいてYを変化させます。XとYの両方を変化させれば、斜めに動くことになります。

ここで注意したいのは、座標の変化する幅はできるだけ小さくしたほうが記号が動いているように見えるということです。一度に10字ぶん移動させるのと、1文字ぶんずつ移動して10文字ぶん移動させるのとでは、結果としては同じ位置に移動しますが、見た目には違った感じになります。そこで本項では、座標を最小単位の1文字ぶんずつ変化させることにします。

また、ある座標の位置に表示した記号は、次の位置に移動する前に消してやる必要があります。この手順を省くと、移動した数だけ記号が表示されてしまうので、記号が動いているようには見えず、ただズラッと並ぶことになります。



現在の位置を●（2, 2）とすると、それぞれの位置に動かすには、座標の値を次のようにすればよい。

A →(2−1, 2−1)＝(1, 1)
B →(2, 2−1)＝(2, 1)
C →(2+1, 2−1)＝(3, 1)
D →(2−1, 2)＝(1, 2)
E →(2+1, 2)＝(3, 2)
F →(2−1, 2+1)＝(1, 3)
G →(2, 2+1)＝(2, 3)
H →(2+1, 2+1)＝(3, 3)

▲表示位置（座標）の移動

くり返し処理はFOR～NEXTが便利

記号を動かすには、次のような手順になります。連続して移動させるには、②～④をくり返せばよいわけです。

①最初の位置を指定する。

②記号を表示する。

③記号を消す。

④座標を変化させて位置を移動する。

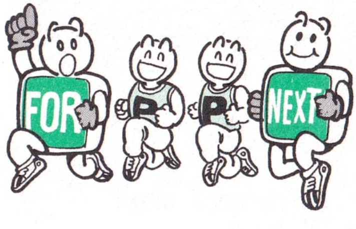
使用する命令は位置指定命令 **LOCATE** と表示命令 **PRINT** の2種類だけですが、これをそのままプログラムにすると、たいへん長いプログラムになることはすぐに予想がつくことと思います。たとえ5文字ぶん動かすにしても、②～④を5回くり返すのですから15行のプログラムになり、これに①の行と、最後に終止命令 **END** を含む行が必要ですから、合計で17行ものプログラムになります。これでは画面全体に記号が動きまわるプログラムをつくるとなると、たいへんな手間がかかります。また、入力する手間も、それだけ多くなります。

当然ですが、②～④のくり返し処理を簡略にすれば、プログラムも短くなります。MSXベーシックには、そのための命令があります。

基本形

FOR␣変数＝初期値␣TO␣終値␣
STEP␣増しぶん
～
NEXT␣変数

FORとNEXTにはさまれた命令を、指定した回数だけくり返すくり返し命令



くり返し処理は、**FOR** と **NEXT** によって命令します。基本形のうち、変数とは自由に値を定められる数のことで、ふつう、アルファベットで示します。数学でA=1、B=2などと書く、AやBが変数です。イコールのあとの数字によって、AやBの値を指定するわけです。これに対して、はじめから値の定まった数を、定数といいます。なお、変数は何文字でもよいのですが、最初の2文字しか判別されません。

この命令を実行すると、変数の値が初期値から増しぶんごとに増えていき、終値になるまで実行をくり返します。

くり返し命令を使ってみよう

くり返し処理は、プログラムの中でたいへんよく使われます。人間には単調で時間のかかるくり返しも、パソコンがすばやくやってのけます。こんなところでも、パソコンの便利さがよくわかります。

簡単な例で、**FOR～NEXT** を使ってみましょう。下のプログラムを入力して、実行してみてください。このプログラムでは、I（変数）の値は1（初期値）から始まり、プログラムの行20の命令を実行し、**NEXT** までくると行10に戻ってIの値を1（増しぶん）増やします。これをくり返して、Iの値が10（終値）になると終わり、行40にいって終了します。

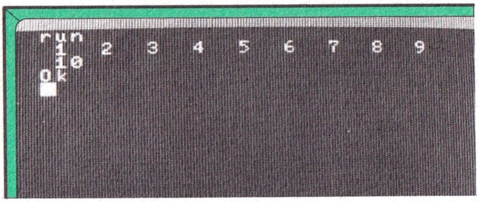
FOR と **NEXT** はつねにペアで使われ、これによるくり返し処理を FOR-NEXT ループといいます。**STEP** の後ろに指定する増しぶんは、負の値（マイナス）でもかまいませんが、その場合には、初期値が終値より大きくなければなりません。また、**FOR** の変数と **NEXT** の変数は同じものでないと、「NEXT without FOR」というエラーメッセージが出ます。

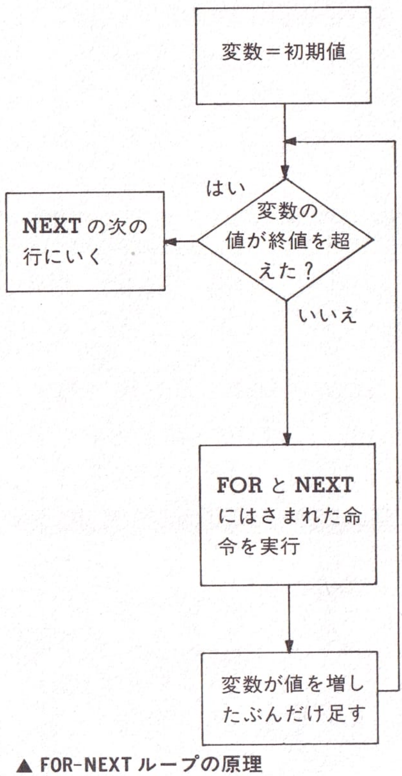
増しぶんが(＋) 1のときには **STEP** が省略でき、**STEP** 以下を入力しないと、増しぶんは自動的に（＋）1になります。

●プログラム

```
10 FOR I=1 TO 10 STEP 1
20     PRINT I;
30 NEXT I
40 END
```

実行





▲ FOR-NEXT ループの原理

同じPRINT(プリント)でも表示方法はいろいろ

表示命令PRINT(プリント)についても使う機会が多いので、ここでまとめておきましょう。先の基本形ではダブルコーテーションがありましたが、もしこれがないと、表示されないのかどうかを調べてみます。また、コンマ(,)やセミコロン(;)の働きも、同じプログラムを実行して確認します。記憶消去指令NEW(ニュー)を実行して、前のプログラムを消してから、次ページのプログラムを入力してください。プログラムとその実行結果を見比べながら、各記号の意味を説明していきましょう。

✣ダブルコーテーション(" ")がないと、文字は表示されない

行10～30と、行50、行60を比べてみると、ダブルコーテーションの有無によって表示結果が違うことがわかります。ダブルコーテーションがなくても表示されるのは、数字だけです。アルファベットはダブルコーテーションがないと変数とみなされますが、この変数の値を決めるプログラムがないので、最初に入っていた0が表示されます。もし別の値が入っていれば、その値が表示されます。また、ダブルコーテーションは、前だけにつけてもその文字が表示されます。

✣コンマ（ , ）はその左右の文字列を13字ぶんあける

行70はコンマの働きを見るための行です。コンマでつないだ2つの文字の集まり（文字列）は、間隔をあけて表示されます。この間隔はつねに一定で、13字ぶんです。

✣PRINT(プリント)だけだと、行を改める

PRINT(プリント)のあとに何も書かないプログラムを実行すると、1行ぶんの空白を表示することになり、結果として改行を行ないます。

✣セミコロン（ ; ）は、その左右を横につないで表示する

文字列をセミコロンでつなぐとどうなるかは、行90～100の実行結果を見ればわかるでしょう。同じ行の中でセミコロンを使った場合でも、前の行の最後にセミコロンをつけて2行にまたがらせて使った場合でも、2つの文字列は空白なく横に続けて表示されます。

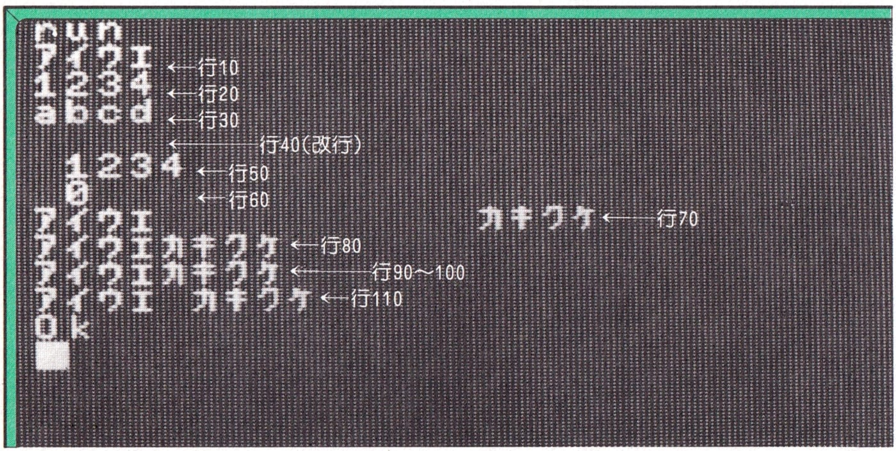
✣ダブルコーテーション内の空白は、そのまま表示される

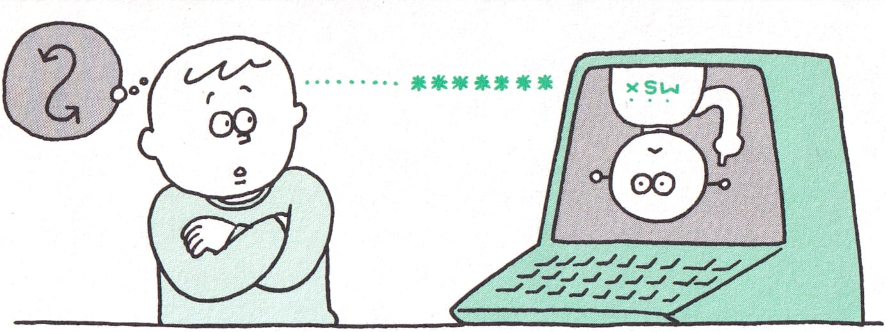
行120では、ダブルコーテーションの中の空白を、そのまま空白として表示されています。「␣アイウエオ」や「アイウエ␣」などのように空白が文字列の前や後ろにある場合でも、同じように空白として表示されます。

●プログラム

```
10 PRINT "アイウエ"            ┐ ダブルコーテーション
20 PRINT "1234"               ┘
30 PRINT "abcd                ← ダブルコーテーションが前だけ
40 PRINT
50 PRINT 1234                 ┐ ダブルコーテーションなし
60 PRINT ABCD                 ┘
70 PRINT "アイウエ","カキクケ"   ← コンマでつなぐ
80 PRINT "アイウエ";"カキクケ"   ┐
90 PRINT "アイウエ";           │ セミコロンでつなぐ
100 PRINT "カキクケ"           ┘
110 PRINT "アイウエ カキクケ"
120 END
```

実行





LESSON ③

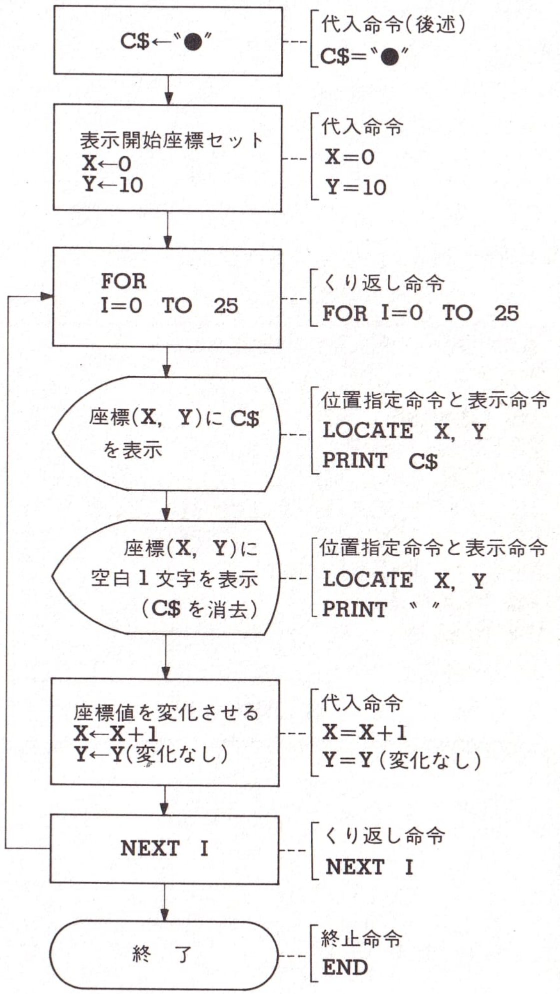
記号を横、縦、斜めに動かす

記号を横に動かすには

ではさっそく、くり返し命令 FOR～NEXT（フォー・ネクスト）を使って「●を（0, 10）から右へ1字ぶんずつ動かして25字ぶん動いたら止まる」という課題にとりくむことにします。処理の手順は、次のようになります。

①最初の座標（X＝0, Y＝10）を設定する。
②座標（X, Y）に●を表示する。
③座標（X, Y）に表示されている●を消す。
④座標Xに1を加える（Yはそのまま）。
⑤以上のうち、②～④を26回くり返す。

くり返し処理が26回となっているのは、最初の位置での表示が1回ぶん含まれるからです。上の手順に少しくふうを加えて、フローチャートをつくります。



記号を横に動かすプログラム

フローチャートをもとに、プログラムをつくります。上から順に命令文にしていき、行番号をつければできあがりです。プログラムのしくみを考える前に、行20、行40〜60に注目してください。ごく短い命令文や同じような内容の命令文は、それだけで1行とるのは手間なので、2つの命令文をつないで1行にします。

複数の命令文をつなぐのには、区切りごとにコロン（：）を使います。たとえばX＝0とY＝10をつなぐ場合には、「X＝0：Y＝10」とすればよいのです。これをマルチステートメント（多重命令）といい、うまく使うとプログラムの行の数を少なくすませることができます。

このプログラムでもっとも重要なところは、行30〜70です。行60で「X＝X＋1：Y＝Y」となっていますから、FOR-NEXT（フォー・ネクスト）ループが1回実行されるたびに、Xの値が1ずつ増えていきます。Yの値は変わりません。

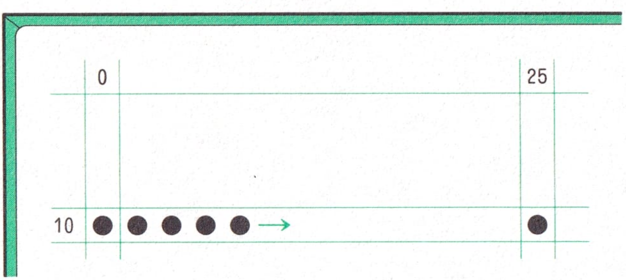
このように水平座標だけが1ずつ増えていくために、記号は右へ1字ぶんずつ移動していくのです。

記号が移動しているように見せるための表示と消去は、行40と行50で行なっています。行50の表示命令**PRINT**（プリント）のダブルコーテーション内には、何もありません。座標の値は行40と同じ（X，Y）ですから、行40の表示がすぐに行50で消されるわけです。記号を左に移動させるには、行20のXに適当な値を入れて、行60を「X＝X−1」にします。

●プログラム：記号を横に移動させる

```
10 C$="●"
20 X=0:Y=10
30 FOR I=0 TO 25
40      LOCATE X,Y:PRINT C$
50      LOCATE X,Y:PRINT " "
60      X=X+1:Y=Y
70 NEXT I
80 END
```

F10 で実行



変数の内容を決める代入命令

行 40 の後半部分（PRINT　C$）が、これまでに説明したのと違っていることに気がついたでしょうか。ダブルコーテーションがありません。このままだとなにも表示されないはずですが、表示の内容は、すでに行 10（C$＝●）で決められているのです。

変数を扱うプログラムでは、このように、変数の内容をあらかじめ指定することができます。これを代入といいます。この例でいえば、行10でパソコンの記憶装置 RAM（ラム）の中に、「C$」という変数名を準備し、そこに「●」という記号を収納します。そして、行 40 の「PRINT　C$」で、C$の内容つまり「●」を画面に表示する、ということになります。

このように、表示命令 **PRINT**（プリント）のあとの変数には、ダブルコーテーションをつけません。もしこれにダブルコーテーションをつけて「PRINT　"C$"」とすると、「●」が表示されず、「C$」という文字がそのまま表示されることになります。代入には、代入命令 **LET**（レット）を使います。

代入命令 **LET**（レット）は、省略してもさしつかえないので、本書ではすべて省略しています。この命令を実行すると、右辺にある数値や文字が、左辺の変数名に代入（収納）されます。ただ、数学でいう代入とは違い、右辺と左辺をとりかえることはできません。ベーシックで使うイコール（＝）は、数学的な意味の等号ではなく、矢印（←）だと考えたほうがわかりやすいでしょう。

たとえば前出のプログラムで、行 60 に「X＝X＋1」という代入文があります。数学上では成立しないはずですが、ベーシックではまちがいではありません。この式は「X←X＋1」という意味ですから、右辺の「X＋1」を左辺の X に代入するということです。このような形の代入式は、FOR-NEXT（フォーネクスト）ループにつきものです。

基本形

(LET（レット）␣)変数名＝数値や文字

変数に数値や文字を代入（収納）する代入命令。LET（レット）は通常、省略する

代入と関係のある変数というものの性質について、ある程度は理解できたでしょうか。変数を細かく分類すると、数値型変数と文字型変数とになります。

数値型変数とは、金額や数量などの数値を代入するもので、代入する数値をそのままイコールの右辺に書きます。

◉変数の種類と使い方

変数の種類	数値型変数	文字型変数
変数名のつけ方	英字、または英字と数字 〈例〉A、AB、A 5、	左の変数名に $（ダラー）記号をつける 〈例〉A$、AB$、A 5$
右辺の書き方	代入するものをそのまま書く 〈例〉A＝5 AB＝207 A 5＝117	代入するものをダブルコーテーションではさむ 〈例〉A$＝"アイ" AB$＝"A" A 5$＝"■"

一方、文字型変数とは、名前や文章などの文字を代入するものです。代入するものを、ダブルコーテーションではさんでイコールの右辺に書きこみます。両者はこのように、変数名のつけ方や右辺の書き方が違います。

変数は、A、B、……、または A$、B$……などと変数名を設定すると同時に、その内容が自動的に決まります。この内容のことを初期値といいます。数値型変数の初期値は 0 で、文字型変数の初期値は空白(ヌルストリング)です。

したがって、先に変数の内容を決めないままで変数を使うと、その変数の値は初期値のまま扱われます。プログラムの中で、変数 A の値を設定せずに「PRINT A」とすると、A の初期値は 0 ですから 0 が表示されます。この変数が A$ というように文字型であれば、空白ですから何も表示されません。

C/O/L/U/M/N

◉よく使う命令は機能キーに設定◉

プログラムによっては、ある特定の命令をよく使う場合がある。絵を描く場合など、直線命令の LINE(ライン) や円命令 CIRCLE(サークル)、彩色命令 PAINT(ペイント) などをくり返し使うことが多い。これらをいちいち入力するのは、めんどうくさい。機能キーに設定してしまうと便利だ。たとえば F1 に LINE(ライン) を設定しておけば、キー操作が 1 回ですむ。

機能キーへの設定には、機能キー定義指令 KEY(キー) を使い、次のようにする。

KEY　数値，"文字列"

（数値：機能キーの番号　"文字列"：設定する文字）

たとえば、機能キー 5 に LINE(ライン) を設定するなら、「KEY　5，"LINE"」とすればよい。最後に入力終了キー RETURN を押せば完了で、あとは、F5 を押せば LINE(ライン) が表示される。

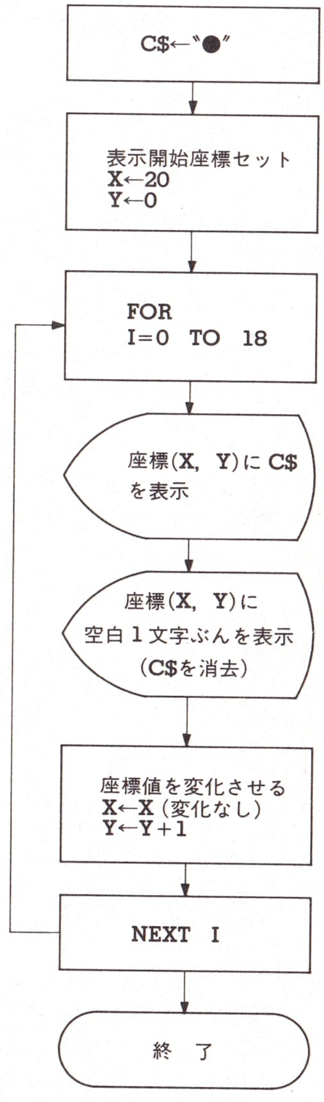
キーの番号は 1 ～10、文字は15文字以内と決められている。ほとんどの命令がセットできるわけだ。

一度定義した内容は、定義しなおすか、電源を切るまでは変わらない。逆にいうと、電源を切れば、機能キーは最初の設定に戻る（32ページ参照）。この機能をうまく使って、効率よくプログラムを入力しよう。

記号を縦に動かすには

記号を縦に動かす手順は、横の場合とほぼ同じです。違うのは、横に動かすときはXの値を変化させてYの値を固定していたのに対し、縦の動きではXを固定し、Yを変化させる点だけです。

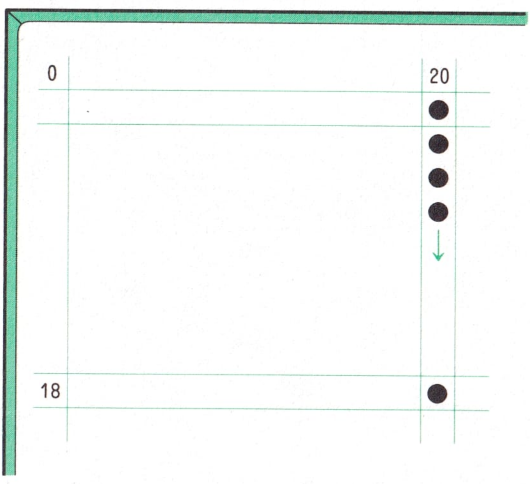
縦に18字ぶん動かす場合、フローチャート、プログラム、実行結果は、それぞれ以下のようになります。



●プログラム：記号を縦に移動させる

```
10 C$="●"
20 X=20:Y=0
30 FOR I=0 TO 18
40      LOCATE X,Y:PRINT C$
50      LOCATE X,Y:PRINT " "
60      X=X:Y=Y+1
70 NEXT I
80 END
```

実行



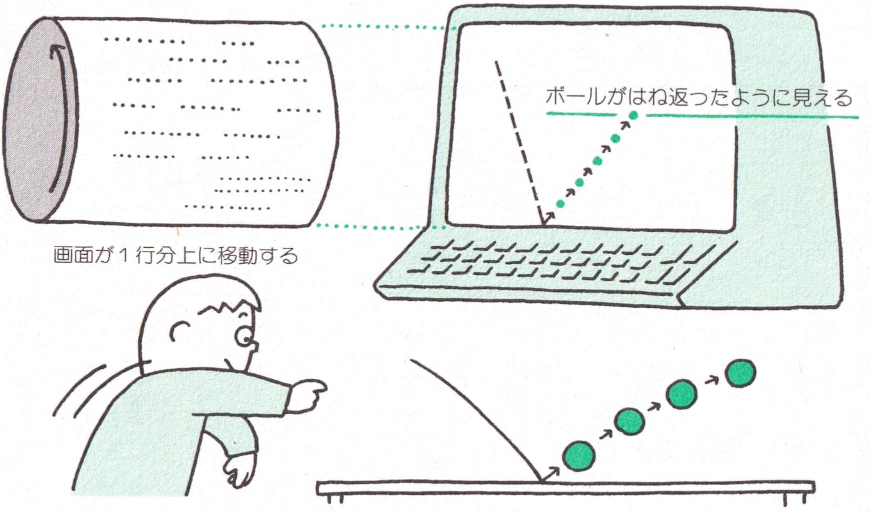
記号の移動範囲が大きすぎると

画面の大きさには限界がありますから、座標の範囲に注意してください。横の移動は最大 0～28、縦は最大 0～23 ですから、その値の範囲を超えると妙なことになります。

横の移動の場合、0 より小さくなると、記号は画面の左端にきたあとで「Illegal function call」というエラーメッセージが出ます。「指定の方法がまちがっているよ」という意味です。逆に、28 を超えると、記号は右端に止まったままで点滅をくり返します。この場合には、エラーメッセージが出ません。

上への移動の場合は、範囲を超えると、やはり「Illegal function call」というエラーメッセージが出ます。おもしろいのは下への動きのときで、下端までくると、次からは上に向かって 1 字ぶんおきに記号が表示されます。ちょうど、落ちてきたボールが地面にはね返ってはずんだように見えます。

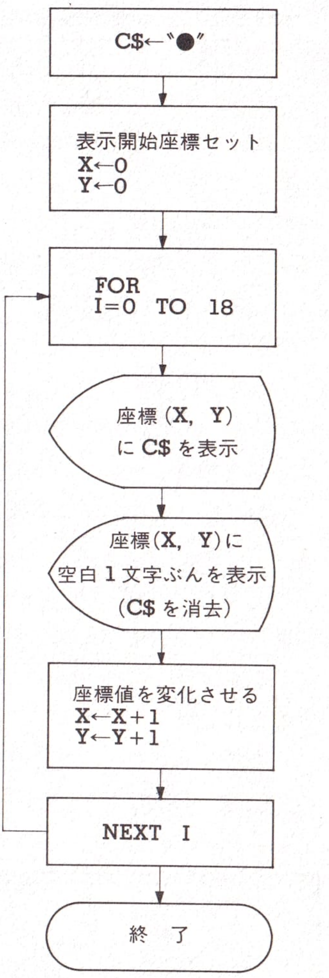
これは、画面いっぱいまできてなお表示する命令があるときには、画面全体が 1 行ぶん上へ持ちあがる働きをするからです。これをスクロールといいます。1 行ぶん持ちあがるごとに表示と消去（空白）が実行されるので、1 字おきに表示されるわけです。



▲スクロール

記号を斜めに動かそう

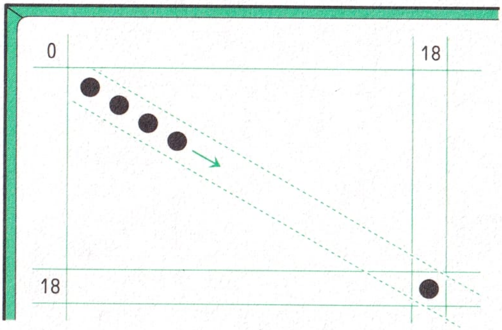
記号を斜めに動かすには、座標のXとYを同時に変化させます。例にあげたのは「●を左上すみ(0, 0)から(18, 18)の位置まで移動させる」という課題で、行60が、いままでのプログラムと違っています。移動の方向を変えるには、この行の「+1」を「−1」にすればよいのです。



●プログラム：記号を斜めに移動させる

```
10 C$="●"
20 X=0 :Y=0
30 FOR I=0 TO 18
40      LOCATE X,Y:PRINT C$
50      LOCATE X,Y:PRINT " "
60      X=X+1:Y=Y+1
70 NEXT I
80 END
```

実行



練習問題4

次の3つの課題を行なうプログラムをつくってみよう。

①●を(0, 17)から(17, 0)まで、1字ぶんずつ移動するプログラム。

②●を(28, 16)から(15, 3)まで1字ぶんずつ移動するプログラム。

③●を右上すみから(11, 17)まで1字ぶんずつ移動するプログラム。

LESSON――④

記号をいろいろに動かす

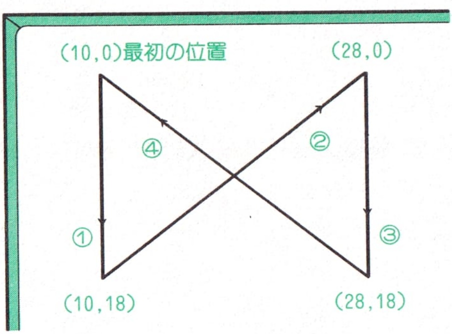
規則的なパターンを描く

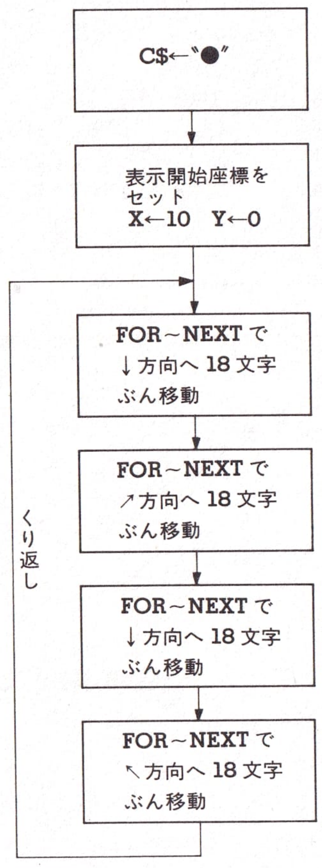
横、縦、斜めの動かし方をマスターすると、これらの組み合わせによって一定のパターンの軌跡を描かせることが可能です。横、縦を組み合わせれば四角形を描きますし、斜めの動きを加えれば対角線も描きます。さしずめマイコンによるひと筆書きといったところでしょう。

下図のように、(10, 0) の位置からスタートして、蝶（ちょう）ネクタイのような模様を描く、というのがここでの課題です。大まかな手順は、次のようになります。

①下の方向へ(10, 18)の位置まで移動する。
②右上の方向へ(28, 0)まで移動する。
③下の方向へ(28, 18)まで移動する。
④左上の方向へ(10, 0)つまり最初の位置まで移動する。

どの場合も、動くのは18字ぶんですから、FOR-NEXT（フォー・ネクスト）ループは、「FOR I=0 TO 17」です。





いつまでもくり返すにはGOTO（ゴーツー）命令で

フローチャートではFOR-NEXT（フォー・ネクスト）ループが4つ使われ、最後のループが終わったところで、最初のループに戻しています。FOR～NEXT（フォー・ネクスト）が指定した回数だけくり返すのに対して、このくり返しは無限に続きます。これを無限ループといい、行230がそれを行なっています。それに使われるのが新しい命令GOTO（ゴーツー）です。

基本形

GOTO（ゴーツー）␣行番号

指定した行番号まで無条件に飛びこす飛びこし命令

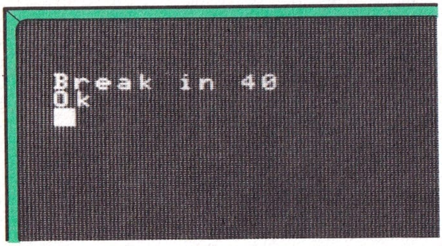
基本形で、GOTO（ゴーツー）に続く行番号によって、ゆく先を指定します。この行番号を、ゆく先番号といいます。プログラムを実行すると、行30～220まで実行したあと、行230の「GOTO 30」によって再び行30へ戻り、記号をいつまでも動かします。

●プログラム

```
10 C$="●"
20 X=10:Y=0              ←最初の位置
30 FOR I=0 TO 18         ┐
40     LOCATE X,Y:PRINT C$│
50     LOCATE X,Y:PRINT " "│ ↓方向の移動
60     X=X:Y=Y+1         ┘
70 NEXT I
80 FOR I=0 TO 18         ┐
90     LOCATE X,Y:PRINT C$│
100    LOCATE X,Y:PRINT " "│ ↗方向の移動
110    X=X+1:Y=Y-1       ┘
120 NEXT I
130 FOR I=0 TO 18        ┐
140    LOCATE X,Y:PRINT C$│
150    LOCATE X,Y:PRINT " "│ ↓方向の移動
160    X=X:Y=Y+1         ┘
170 NEXT I
180 FOR I=0 TO 18        ┐
190    LOCATE X,Y:PRINT C$│
200    LOCATE X,Y:PRINT " "│ ↖方向の移動
210    X=X-1:Y=Y-1       ┘
220 NEXT I
230 GOTO 30              ←行30へ（無限のくり返し）
```

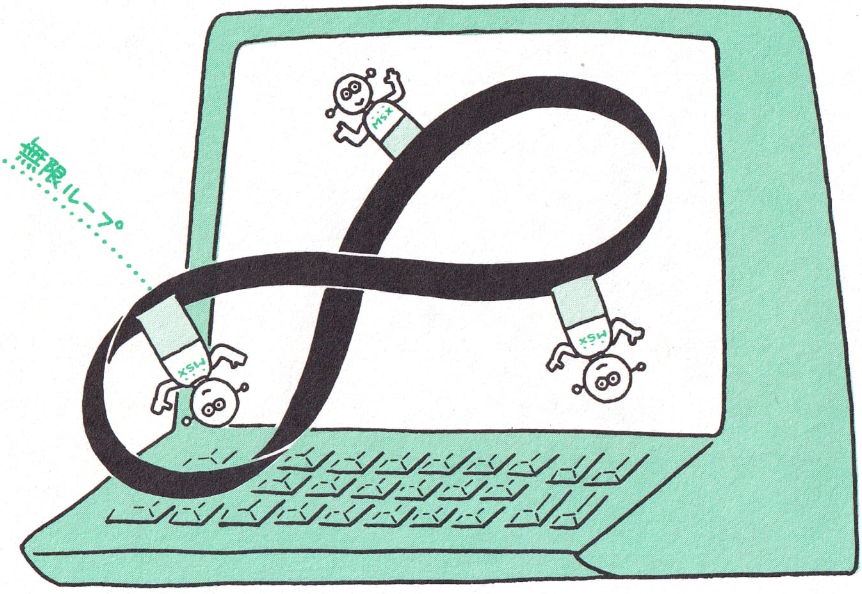
無限ループを終わらせるには

無限ループを終わらせるには、CTRL を押しながら STOP キーを押します。ブザーが鳴って、次のようなメッセージが画面にあらわれます。



「Break in 40」（ブレイク イン）は、行 40 でプログラムの実行が中断されたことを意味します。この数字は、STOP キーを押すタイミングによって異なります。その下には「Ok」と出てカーソルが点滅していますから、命令が入力されるのを待っている状態であることを示しています。そこで実行指令 **RUN**（ラン）を入力すれば、再び最初からプログラムが実行されます。

また、無限ループを一時停止させるには、STOP キーだけを押します。この場合は、とくにメッセージが出ません。STOP キーを押したところで記号の動きは止まります。もう一度 STOP キーを押して再開すると、その位置から再び動きはじめます。



でたらめに動かすには RND(ランダム) 関数

記号の動きをもっと複雑にするには、一定の方向に動く距離を短くするようにくふうします。少し動いては方向を変え、その方向で少し動いてはまた方向を変える、ということをくり返してやればよいのです。

また、斜めへの動きは、これまで水平方向、垂直方向とも1文字ぶんずつ変化させましたが、これを2文字ぶん、3文字ぶんと変えると、進む角度が変化して動きはさらに複雑になります。

それでも結局は同じ処理のくり返しですから、どうしても規則的な動きになります。予測がつかないようなでたらめな動きにするには、どうしたらよいでしょうか。それには、水平座標Xと垂直座標Yにでたらめな数字を代入して、記号が表示・消去される位置を不規則にすることです。

規則性がなく、どの数も同じ割合で発生するような数を乱数といいます。ベーシックでは、乱数を発生させるために **RND**(ランダム) を使います。これは命令ではなく、関数です。関数とは、ひとつの条件を与えると、それによってひとつの値が決まる性質を持つものです。

基本形のうち、「X」は、数式や数値、数値型変数のことをあらわします。その例は、次のとおりです。

①RND(1)——0以上1未満の乱数を発生させる。これをもとにさまざまな加工を加えることで、必要な乱数を発生させることができる。

②RND(1)+1——1と2のあいだの乱数（2は含まない）。

③RND(1)*5——0と5のあいだの乱数（5は含まない）。*は乗算の記号。

下のプログラムを実行すると、乱数が20個発生します。表示命令 **PRINT**(プリント) のあとの「RND(1)」を、RND(1)+1、RND(1)*5などに変えて実行してみましょう。

基本形

RND(ランダム) (X)

与えられた条件で乱数を発生させる乱数関数

●プログラム

```
10 FOR I=1 TO 20
20     PRINT RND(1)
30 NEXT I
40 END
```

INT関数ですべて整数に

すでに見たように、乱数関数**RND**で発生する乱数には、小数点以下のついた値も含まれています。しかし、座標を指定するには整数でなければならないので、すべて整数にする必要があります。その働きをするのが整数関数**INT**です。基本形の（X）には、数式や数値変数が入ります。そこで、乱数関数で発生した値を整数変数のXに代入すれば、整数の乱数を発生させることができます。たとえば、

X＝RND(1)＊5…………①　　　INT(X)…………②

とすれば、②で得られるのは0～4の乱数です。つまり、0、1、2、3、4 が不規則に発生することになります。一般にはこの①②をまとめて、「INT(RND(1)＊5)」と書きます。

基本形

INT　(X)

与えられた数値を超えない最大の整数を求める整数関数

本論に戻って、画面いっぱいを使った座標（X, Y）に代入する座標を考えてみましょう。Xは27まで、Yは21までの乱数にすればよいのです。

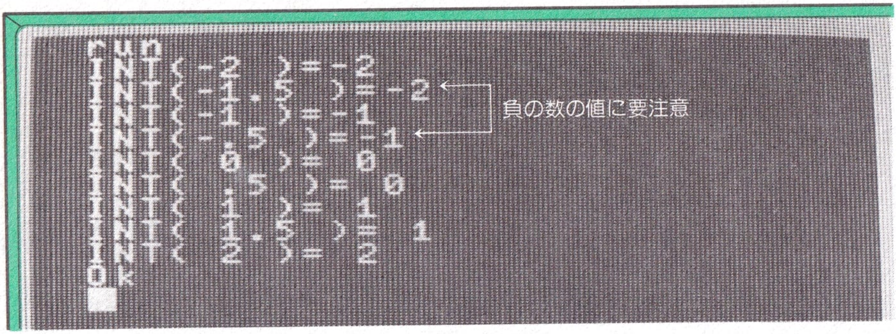
X＝INT(RND(1)＊28)

Y＝INT(RND(1)＊22)

●プログラム

```
10 FOR I=-2 TO 2 STEP .5
20 PRINT "INT(";I;")=";INT(I)
30 NEXT I
40 END
```

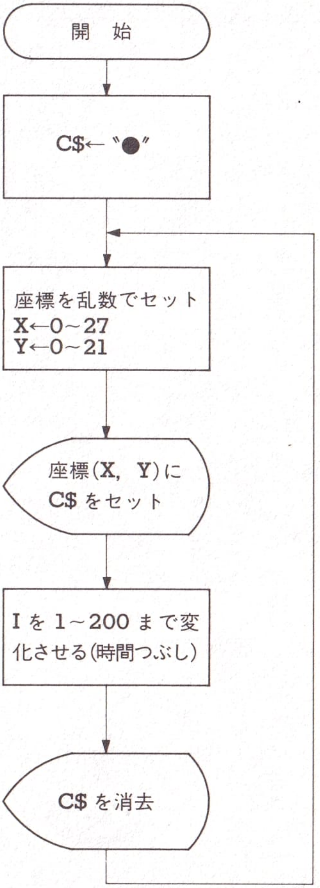
実行



記号がでたらめに動くプログラム

「●をでたらめな位置に表示する」という課題を行なうフローチャートとプログラムは、以下のようになります。ここで「時間つぶし」という処理が入っています。これは、一度表示した「●」を、しばらく表示しておくためです。コンピュータの処理速度はたいへん速いので、この時間つぶしをしないとアッというまに消えてしまい、見た目におもしろみがないのです。

時間つぶしには、くり返し命令を使います。FORとNEXTのあいだにほかの命令文をはさまなければ、FOR-NEXTループがグルグルまわることになり、そのあいだはすぐ前の行が実行された状態で停止しています（行50～60）。TOの次にある終値の値をいろいろ変えて実行してみると、どの程度の終値にすれば、どのくらい時間つぶしができるかがわかります。



●プログラム

```
10 C$="●"
20 X=INT(RND(1)*28)
30 Y=INT(RND(1)*22)
40 LOCATE X,Y:PRINT C$
50 FOR I=1 TO 200
60 NEXT I
70 LOCATE X,Y:PRINT " "
80 GOTO 20
```

練習問題 5

次の2つの課題を実行するプログラムを考えてみよう。

①水平座標10の一線上に●をでたらめに表示する。

②垂直座標15の一線上に●をでたらめに表示する。

LESSON ⑤

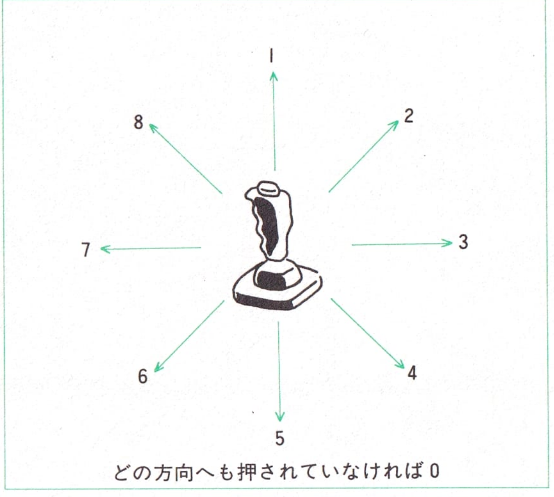
ジョイスティックを自由に扱おう

ジョイスティックを使うとゲーム気分

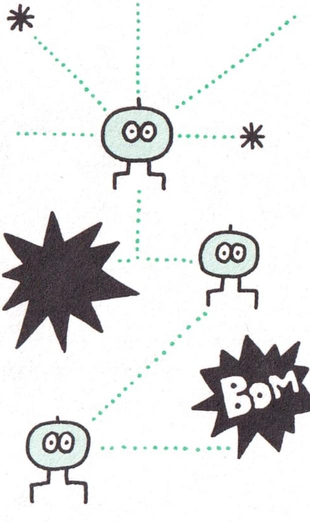
MSX パソコンには、ジョイスティックを接続する場所があります。ここでは、ジョイスティックを使って自由自在に記号をあやつるプログラムをつくってみることにしましょう。プログラムづくりにとりかかる前に、なぜジョイスティックで記号が動かせるのか、そのしくみを簡単に説明します。スティックを左へ倒して、記号を左へ動かす場合で考えてみます。

①ジョイスティックを左へ倒す。
②パソコンが、スティックの傾いた方向を判断する。
③記号を左へ動かす。

問題になるのは②です。スティックの方向をパソコンが判断できるのは、スティックの傾いた方向を 8 つの方向に分類し、それぞれを数字におきかえているからです。ですから、プログラムでこの値を読みとり、それぞれの値に合った方向へ進むようにしてやればよいのです。



▲ジョイスティックの方向とその値



スティックの傾きを読みとるスティック関数STICK

ジョイスティックがどの方向に傾いているかを調べ、それを数値（関数の値）におきかえるのがスティック関数**STICK**です。基本形の(X)は、どのジョイスティックを使うかで異なり、ジョイスティックをポート1につないだときにはXを1、ポート2につないだときにはXを2にします。また、Xを0にすると、ジョイスティックのかわりにカーソル・キーを使うこともできます。そのときの関数の値(ジョイスティックの方向）と使うキーの関係は次のとおりです。どの方向へも押されていなければ、関数の値は0になります。

↑…1（カーソル・キーでは↑）↗…2（↑→）→…3（→）↘…4（→↓）
↓…5（↓）↙…6（↓←）←…7（←　↖…8（←↑）

基本形

STICK (X)

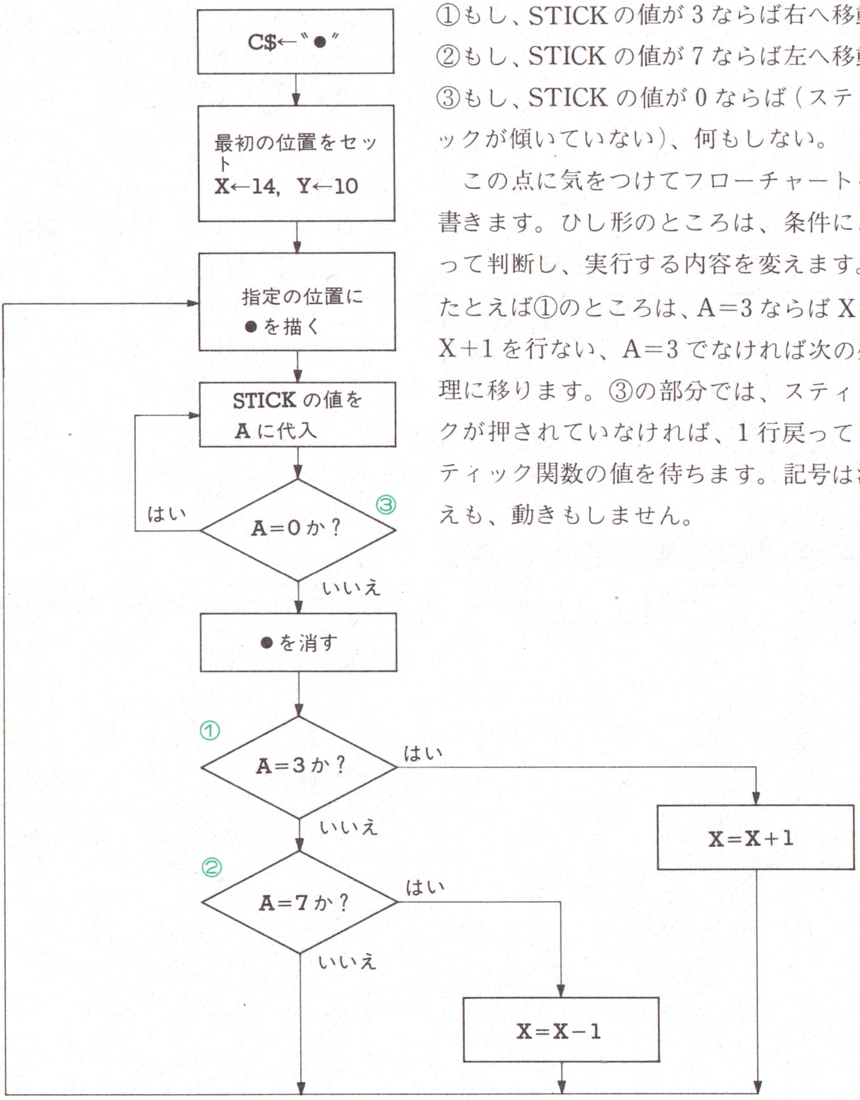
ジョイスティックがどの方向に傾いているかを調べるスティック関数

ジョイスティックをポート1につないだ場合(STICK(1))でも、ポート2につないだ場合(STICK(2))でも、あるいはカーソル・キーを使う場合(STICK(0))でも、方向とスティック関数の値との関係は同じです。これらを、以下の表にまとめておきます。

ジョイスティックの傾き	STICK関数の値	プログラム上での処理(座標の移動)	記号の動き
↑	1	X=X：Y=Y−1	↑
↗	2	X=X+1：Y=Y−1	↗
→	3	X=X+1：Y=Y	→
↘	4	X=X+1：Y=Y+1	↘
↓	5	X=X：Y=Y+1	↓
↙	6	X=X−1：Y=Y+1	↙
←	7	X=X−1：Y=Y	←
↖	8	X=X−1：Y−1	↖

ジョイスティックで記号を横に動かすには

ジョイスティックを使って記号を横に動かすプログラムをつくってみましょう。ここでかんじんなのは、STICK（スティック）関数と処理のつながりです。



①もし、STICK の値が 3 ならば右へ移動。
②もし、STICK の値が 7 ならば左へ移動。
③もし、STICK の値が 0 ならば（スティックが傾いていない）、何もしない。

この点に気をつけてフローチャートを書きます。ひし形のところは、条件によって判断し、実行する内容を変えます。たとえば①のところは、A=3 ならば X=X+1 を行ない、A=3 でなければ次の処理に移ります。③の部分では、スティックが押されていなければ、1 行戻ってスティック関数の値を待ちます。記号は消えも、動きもしません。

スティックで記号を縦、横に動かすプログラム

フローチャートをもとにして、プログラムをつくります。行 50 と行 70〜80 に新しい命令が出てきました。ここは、フローチャートでひし形にあたる部分です。使われている命令は、判断命令 **IF〜THEN** です。**IF** の次にある条件（A＝0 など）によって、**THEN** 以下の命令文を実行します。条件が満たされていれば **THEN** 以下を実行し、そうでなければ **THEN** 以下の命令文を実行しないで、すぐ下の行の実行に移ります（くわしくは 117 ページ参照）。

プログラムを入力して、実行し、ジョイスティックで遊んでください。左右の方向以外にどんな動きもしないのは、上下や斜めのことはプログラム中に入っていないからです。

「Y＝Y」という代入文がないのに気がつきましたか。くり返しているあいだ、変数は、新たな値が代入されないかぎり同じ値ですから、とくに必要ありません。

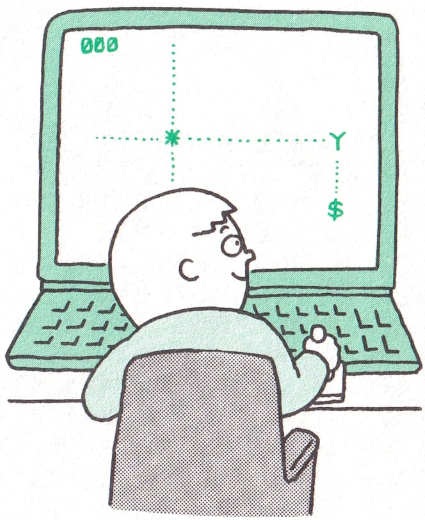
縦にも動かすには、2 行を追加すればすみます（下のプログラム）。

●プログラム

```
10 C$="●"
20 X=14:Y=10
30 LOCATE X,Y:PRINT C$
40 A=STICK(1)
50 IF A=0 THEN GOTO 40
60 LOCATE X,Y:PRINT " "
70 IF A=3 THEN X=X+1
80 IF A=7 THEN X=X-1
90 GOTO 30
```

●プログラム

```
10 C$="●"
20 X=14:Y=10
30 LOCATE X,Y:PRINT C$
40 A=STICK(1)
50 IF A=0 THEN GOTO 40
60 LOCATE X,Y:PRINT " "
70 IF A=3 THEN X=X+1
80 IF A=7 THEN X=X-1
90 IF A=1 THEN Y=Y-1
100 IF A=5 THEN Y=Y+1
110 GOTO 30
```



記号を自由に動かそう

縦と横の動きに加えて、斜めにも動かせるようにすると、記号の動きも自由になり、応用がききます。87ページの表を見ながら、前のプログラムに斜めの動きを加えてみましょう。下のプログラムで、行110〜140が、新たに加わった部分です。ここで斜めの動きを行なっています。

ジョイスティックがない場合でも、同じように楽しめます。行40のSTICK(1)をSTICK(0)とすれば、キーボードのカーソル・キーで、ジョイスティックと同じ操作ができます。試してみてください。

なお、ジョイスティックで操作する場合も、前に述べた範囲指定が大きすぎたときと同じような現象が起こるので注意しましょう。上にあげすぎたり、左へ移動させすぎたりすると「Illegal function call」(イリーガル ファンクション コール)のエラーメッセージが出ます。下にさげすぎると、画面がスクロールしてとびとびに記号が並びます。右へ移動させすぎると、右端で点滅します。

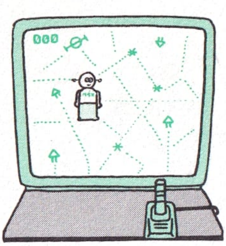
これを避けるには、プログラム上で記号の動く範囲を設定してしまえばよいのです。

どのプログラムも無限ループになっています。プログラムの実行を終わらせるときには、[CTRL] を押しながら [STOP] を押します。一時停止ならば、[STOP] だけを押します。

●プログラム：記号を縦、横、斜めに動かす

```
10 C$="●"
20 X=14:Y=10
30 LOCATE X,Y:PRINT C$
40 A=STICK(1)
50 IF A=0 THEN GOTO 40
60 LOCATE X,Y:PRINT " "
70 IF A=3 THEN X=X+1
80 IF A=7 THEN X=X-1
90 IF A=1 THEN Y=Y-1
100 IF A=5 THEN Y=Y+1
110 IF A=2 THEN X=X+1:Y=Y-1
120 IF A=4 THEN X=X+1:Y=Y+1
130 IF A=6 THEN X=X-1:Y=Y+1
140 IF A=8 THEN X=X-1:Y=Y-1
150 GOTO 30
```

最後に、ちょっとしたいたずらをしてみましょう。下のプログラムの行40「A=STICK(1)」を「A=STICK(1)+1」として実行してみてください。ちょっとおもしろいことが起こります。



·PART·

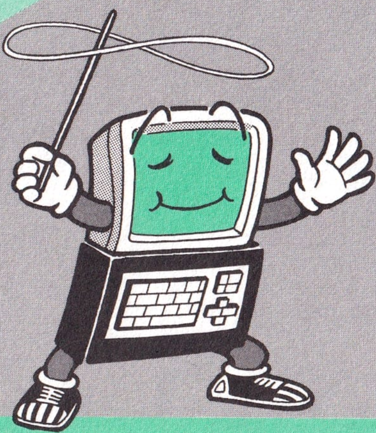
5





うっとり名演奏

パソコン・ミュージック



コンピュータで音楽を楽しむ——MSX 2 パソコンの世界はまだまだ広がります。家庭用テレビや音響装置をつないで名曲を演奏したり、自分で作曲した曲を流したり、楽しみ方はいろいろです。音楽命令自体は複雑ではありません。根気よく手間をかけると、すばらしいものができあがります。この章では、**PLAY**（プレイ）、**SOUND**（サウンド）という命令を学びます。

LESSON—①

まずドレミファソラシド

音楽を演奏するには PLAY 命令

なにはともあれ、MSX 2 パソコンがほんとうに音楽を演奏するのか、試してみることにしましょう。次のようにキーを打ちこんで、最後に RETURN キーを押します。大文字でも小文字でも、結果は同じです。

```
PLAY "CDEFGAB O5C"
```

← RETURN を押す
ダブルコーテーション

ドから次のドまで、1 オクターブが聞こえてきたでしょう。「play」が音楽を演奏する命令、その後ろに続いているのが、音程やオクターブの指定です。基本形のうち、音の指定は、ダブルコーテーションではさむことを忘れないようにしてください。
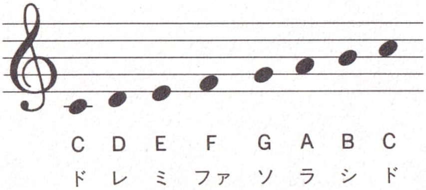
音の指定は、音程やオクターブはもちろん、長さ、大きさ、テンポ、さらに半音の上下、休止符などもできます。これだけできれば、どんな曲でもりっぱに演奏できます。

画面を見ればわかると思いますが、音程の指定は、C、D、E、F、G、A、B の 7 つのアルファベットを使います。「PLAY␣"C"」ならド、「PLAY␣"CEG"」とすればドミソと音が出ます。

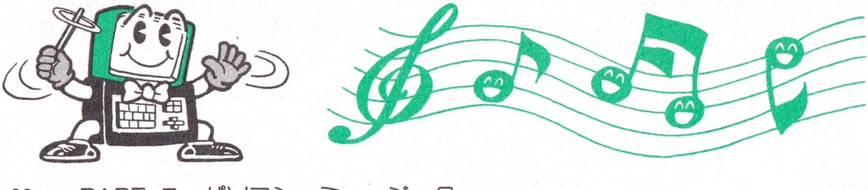
基本形

PLAY␣"音の指定"

指定されたとおりに音楽を演奏する音楽命令



▲音階とアルファベット



オクターブや音の大きさを変える

「PLAY "CDEFGABO5C"」の O5 は、オクターブの指定です。オクターブは、O と、1～8の整数で指定します。電源投入後は O4 になっていますから、O5 とすれば1オクターブ上、O3 とすれば1オクターブ下になります。オクターブの指定は、次の新しいオクターブ指定をするまで有効です。次のようにキーインしてみましょう。1オクターブ低い音になります。

```
PLAY "O3 CDEFGAB O4C"   ← RETURN を押す
```

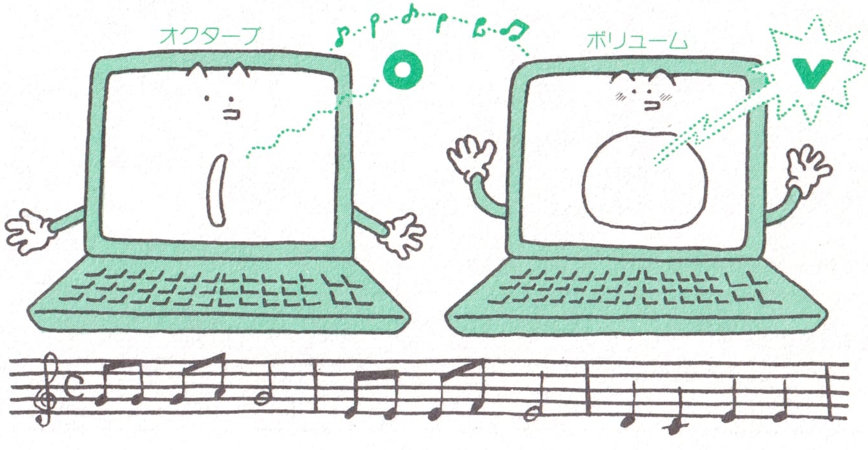
次に、音の大きさを変えてみましょう。

```
PLAY "O3 CDEFG V10 AB O4C"   ← RETURN を押す
```

ドレミファソまできて、ラシドのところで急に音が大きくなることがわかります。これは音の大きさの指定 V10 が入っているからです。V のあとの数をいろいろに変えて試してください。V のあとの数値が大きいほど、音が大きくなります。

ただし、V のあとの数値は、0～15の整数しか使えません。電源投入後は V8 になっています。また、この指定も、新しく大きさの指定をするまで有効です。

なお、O は Octave の頭文字、V は Volume の頭文字をとった記号です。MSX パソコンでは、以上説明した音程のアルファベット、それに O や V など、音に関する指定に使う記号や数値をミュージック・マクロ命令と呼んでいます。



①まずドレミファソラシド

L（エル）で長さを変え、R（アール）でひと休み

音の長さもいろいろあります。楽譜を見ても、2分音符、4分音符、8分音符と長さの違う音符があります。MSXパソコンではどうするか、次の例で試してみましょう。

```
PLAY  "L2 O3 CDEFGAB O4C"  ← RETURN を押す
PLAY  "L8 O3 CDEFGAB O4C"  ← RETURN を押す
```

音の長さは、L（エル）と数値（1～64の整数）で指定します。

𝅝	𝅗𝅥	♩	♪	𝅘𝅥𝅯	𝅘𝅥𝅰	𝅘𝅥𝅱
L1	L2	L4	L8	L16	L32	L64
全音符	2分音符	4分音符	8分音符	16分音符	32分音符	64分音符

この指定も、次の指定まで有効ですが、特定の音だけ長さを指定する場合は、アルファベットのあとに数値を入れます（数値と長さの関係は上と同じ）。

```
PLAY  "L8 O3 CDE 4 FGAB O4C"  ← RETURN を押す
       └8分音符     └数値の前の音(E)だけ4分音符
```

楽譜をプログラムにするときには、この2つの指定方法をうまく使い分けます。楽譜の中に同じ長さの音がいくつもあれば、その音についてはまとめてL（エル）と数値、その他の音については数値だけで指定するようにすると便利です。

付点音符は、ピリオド（ . ）を音程や長さ指定の数値の後ろにつけます。たとえば、「C4.」とすれば付点4分音符のドです。また、半音あげるには♯、または＋、半音下げるには－の記号を、音程の後ろにつけます。

休止符は、R（アール）と数値で指定します。

```
PLAY  "CDEF R4 GAB"  ← RETURN を押す
```

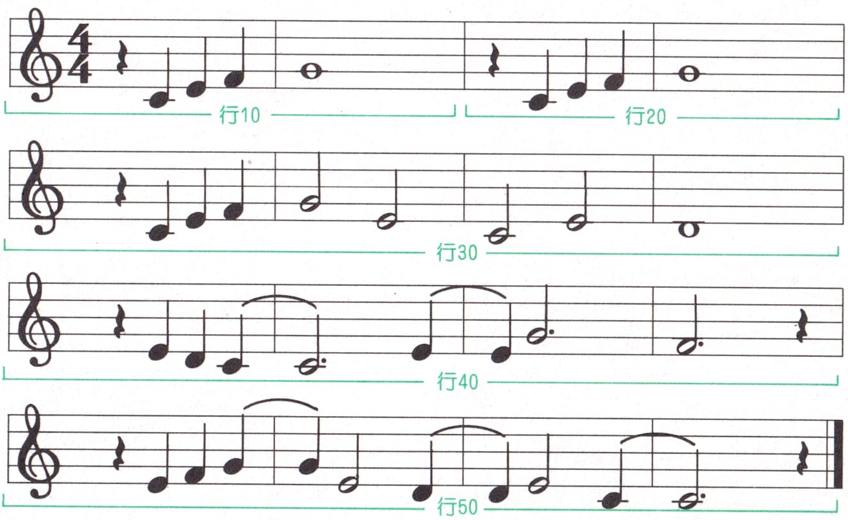
数値と休止符の長さは、下のようになっています。

𝄻	𝄼	𝄽	𝄾	𝄿	𝅀	𝅁
全休止符	半休止符	4分休止符	8分休止符	16分休止符	32分休止符	64分休止符
R1	R2	R4	R8	R16	R32	R64

パソコンで作曲ができる?

短い曲をプログラムにしてみよう

これまで説明した方法で、どんな音楽でもプログラムにできます。さっそく短い曲をプログラムにしてみましょう。



「聖者が街にやってくる」の楽譜です。音の長さや付点、休止符に注意してプログラムにしてください。2〜4小節ずつに分けてプログラムにすると、わかりやすいでしょう。

●プログラム:「聖者が街にやってくる」

```
10 PLAY "L4R4CEFG1"
20 PLAY "L4R4CEFG1"
30 PLAY "L4R4CEFL2GECED1"
40 PLAY "L4R4EDCC2.EEG2.F2.R4"
50 PLAY "L4R4EFGGE2DDE2CC2.R4"
60 END
```

指定を変えるとオリジナル曲に

「聖者が街にやってくる」を十分に楽しんだら、こんどはこのプログラムをもとにして、君だけのオリジナル曲をつくってみることにしましょう。長さを変えたり、適当に音をはさんだり、音を入れかえたり、といろいろ方法はあります。

ここでは、次のような加工をしています（上のプログラム）。

①行 40〜50 の音の長さの指定を変える。

L 4→L 8

②行 10〜30 の音程を変える。

CEF→FEC

実行してみてください。これだけの加工でずいぶん感じが変わります。同じようにしてどんどん変えていけば、いつのまにか作曲ができてしまいます。

曲のテンポを変えても、おもしろい感じになります。MSX パソコンでは何も指定しなければ、4 分音符を 1 分間に 120 回演奏するテンポ（♩＝120）になっています。テンポを変えるのは T（ティー）と数値（32〜255）を使い、たとえば「PLAY "T 200"」という形で指定します。数値が大きいほど速くなります。T（ティー）は Tempo（テンポ）の頭文字です（下のプログラム）。

●プログラム：曲をアレンジする

```
10 PLAY "L4R4FECG1"
20 PLAY "L4R4FECG1"
30 PLAY "L4R4FECGECED1"
40 PLAY "L8R4ED8C2.CE2G2.F2.R4"
50 PLAY "L8R4EFGGE2DDE2CC2.R4"
60 END
```

●プログラム：テンポを変える

```
10 PLAY "T200"
20 PLAY "L4R4CEFG1"
30 PLAY "L4R4CEFG1"
40 PLAY "L4R4CEFL2GECED1"
50 PLAY "L4R4EDC2.CE2G2.F2.R4"
60 PLAY "L4R4EFGGE2DDE2CC2.R4"
70 END
```

LESSON ③

曲に伴奏をつけよう

こうすれば和音が出せる

メロディに伴奏をつければ、もっと本格的に音楽が楽しめます。伴奏をつけるには、同時に複数の音が出せなくてはなりません。MSX パソコンにそれができるかどうか試してみましょう。

次のようにして、入力終了キーを押してみてください。

```
PLAY "C", "E"
```
← RETURN を押す

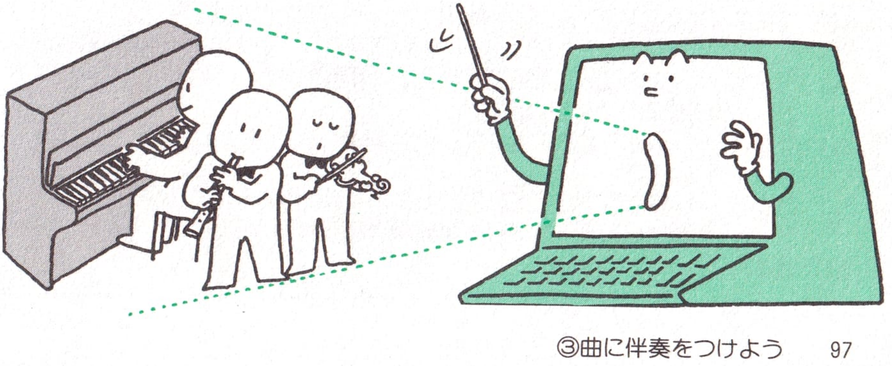
ドとミが同時に聞こえました。ダブルコーテーション（" "）ではさんだ部分を変えれば、自由に 2 重奏ができます。では、3 重奏はどうでしょうか。試してみましょう。

```
PLAY "C", "E", "G"
```
← RETURN を押す

ドミソの和音ができました。これで、3 重奏もできることがわかりました。ついでに 4 重奏も試してみます。
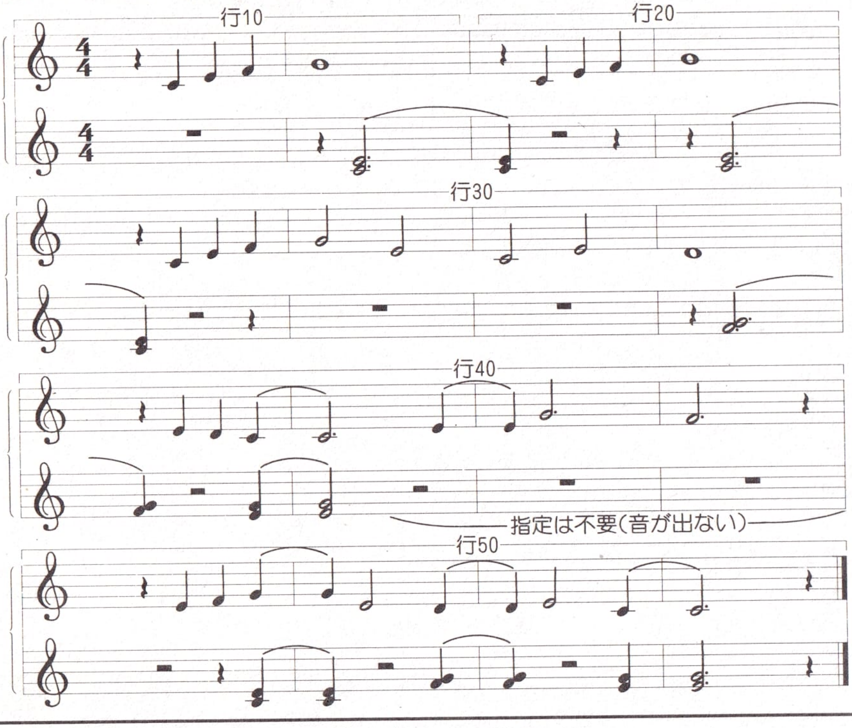
```
PLAY "C", "E", "G", "A"
```
← RETURN を押す

ビーッと音が鳴って、「Syntax Error（シンタックス エラー）」というエラーメッセージが出たでしょう。これで、MSX パソコンで演奏できるのは、3 重奏までだということがわかります。



伴奏をつけてパソコン・ミュージック

先の「聖者が街にやってくる」に伴奏をつけます。どのように伴奏をつけるかは、あらかじめ楽譜にしておくと、プログラムをつくりやすくなります。音を出さないところは、休止符を使います。
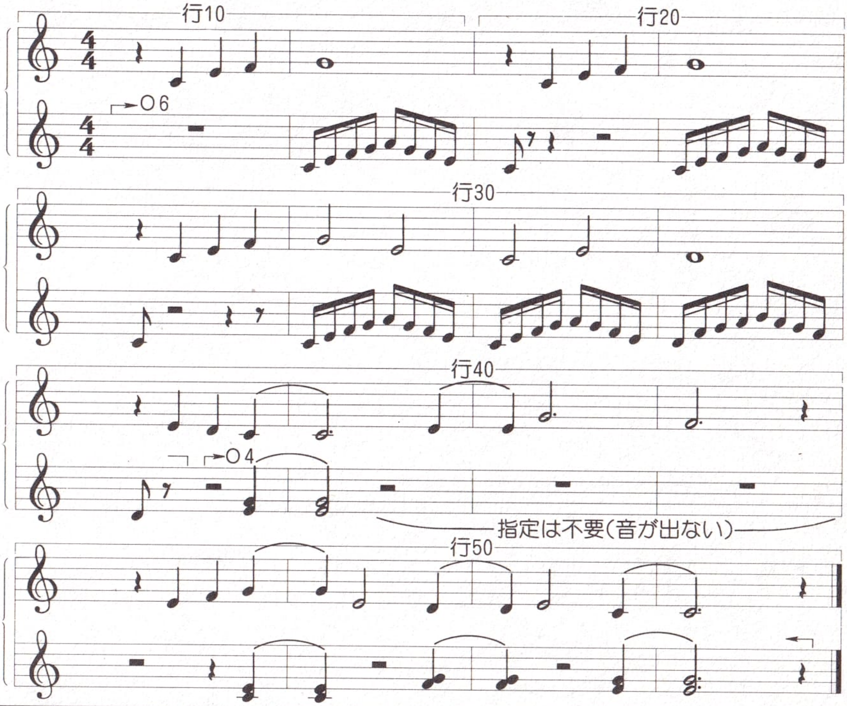


●プログラム：簡単な伴奏つきの「聖者が街にやってくる」

```
10 PLAY "L4R4CEFG1","R1R4C2.","R1R4E2."
20 PLAY "L4R4CEFG1","C4R2R4R4C2.","E4R2R4R4E2."
30 PLAY "L4R4CEFL2GECED1","C4R2R4R1R1R4F2.","E4
        R2R4R1R1R4G2."
40 PLAY "L4R4EDCC2.E2G2.F2.R4","F4R2E4E2","G4R2
        G4G2"
50 PLAY "L4R4EFGGE2DDE2CC2.R4","R2R4C4C4R2F4F4R
        2E4E2.R4","R2R4E4E4R2G4G4R2G4G2.R4"
60 END
```

伴奏をくふうしてみると

もう少し伴奏を複雑にしてみましょう。次のようにすると、さらに本物の演奏らしくなります。かなりめんどうですから、リスト命令 **LIST** で前のプログラムのリストを出して、伴奏の部分だけなおすようにすると、少しは楽にできます。



●プログラム：伴奏つき「聖者が街にやってくる」

```
10 PLAY "L4R4CEFG1","R1O6L8CEFGAGFE"
20 PLAY "L4R4CEFG1","L8CR4.R2CEFGAGFE"
30 PLAY "L4R4CEFL2GECED1","L8CR2R4.CEFGAGFECEF
         GAGFEDFGABAGF"
40 PLAY "L4R4EDCC2.E2G2.F2.R4","D8R8R2O4E4E2",
         "R2R4G4G2"
50 PLAY "L4R4EFGGE2DDE2CC2.R4","R2R4C4C4R2F4F
         4R2E4E2.R4","R2R4E4E4R2G4G4R2G4G2.R 4"
60 END
```

LESSON ④

パソコンで音をつくる

爆弾の音、機関車の音

MSX パソコンは、擬音を自由につくることができます。宇宙船の動く音やビーム砲を発射した音など、ゲームの効果音として使えるような音をつくることもできます。たとえば、こんな音はどうでしょう。次のプログラムを入力して、実行してください。爆弾が落ちるときの音に聞こえませんか。

●プログラム：機関車の音

```
10 SOUND 6,11
20 SOUND 7,7
30 SOUND 8,16
40 SOUND 12,3
50 SOUND 11,114
60 SOUND 13,8
70 END
```

●プログラム：爆弾の音

```
10 FOR I=48 TO 255
20    SOUND 7,254:SOUND
      8,14
30    SOUND 0,I:SOUND 1,0
40 NEXT I
50 SOUND 8,0
60 END
```

基本形

SOUND␣数値①, 数値②

指定どおりにいろいろな音を出す音命令

このプログラムでいくつも使われている SOUND が、音をつくる音命令です。この命令を使うと、音を発生するしくみを直接、コントロールすることができます。基本形のうち、数値①はレジスタ番号といい、0～15 の整数が使えます。数値②はデータといい、数値①がどのような数値になるかによって、指定できる範囲が違います。



簡単にすきな音を探すには

音命令で思うとおりに音をつくるのは、たいへん手間がかかります。レジスタ番号とデータの数値、それに**SOUND**(サウンド)文の組み合わせをいろいろに変える。それに従って音が変わるのですが、その対応のしかたは非常に複雑です。最初に出したい音を考えてプログラムをつくることが、非常にむずかしいのです。

そこで、音探しのプログラムを紹介しておきます。このプログラムを実行すると、ある種の音が出ます。最初に出る音は一定していて、何もしなければ同じ音が続きます。違う音を出したければ、キーボードからどれかのキーを押してください。キーを押したとたんに、違う音が出ます。これをくり返していると、次々に違う音が出てきます。画面には、違う音が出るたびにレジスタ番号の0～7と11～13の値が表示されます。いろいろな音を出してみて、気に入った音が出たら、その値をメモしておきます。それが、**SOUND**(サウンド)文のデータです。

●プログラム：おもしろい効果音を探す

```
10 A=INT(RND(1)*255+1)
20 B=INT(RND(1)*15+1)
30 C=INT(RND(1)*255+1)
40 D=INT(RND(1)*15+1)
50 E=INT(RND(1)*255+1)
60 F=INT(RND(1)*15+1)
70 G=INT(RND(1)*255+1)
80 H=INT(RND(1)*255+1)
90 I=INT(RND(1)*255+1)
100 J=INT(RND(1)*255+1)
110 K=INT(RND(1)*15+1)
120 SOUND 0,A:PRINT "0=";A
130 SOUND 1,B:PRINT "1=";B
140 SOUND 2,C:PRINT "2=";C
150 SOUND 3,D:PRINT "3=";D
160 SOUND 4,E:PRINT "4=";E
170 SOUND 5,F:PRINT "5=";F
180 SOUND 6,G:PRINT "6=";G
190 SOUND 7,H:PRINT "7=";H
200 SOUND 8,31
210 SOUND 9,31
220 SOUND 10,31
230 SOUND 12,I:PRINT "12=";I
240 SOUND 11,J:PRINT "11=";J
250 SOUND 13,K:PRINT "13=";K
260 INPUT X$:CLS
270 GOTO 10
```

気に入った音がいくつか出てきたら、CTRL を押しながら STOP を押してプログラムの実行を中断しましょう。行120～250の部分を入力しなおして、メモした値を**SOUND**(サウンド)文の数値②にあてはめればオーケーです。

ゲームで一服

逃げろや、逃げろ

インベーダーが追いかけてきた。つかまったら、おしまいだ。バリケードを落としながら、どんどん逃げよう。

このプログラムは、ジョイスティックを使うようになっているが、行380、行870の「STRIG」、行480の「STICK」に続くカッコ内の数字を0にし、行360の「GOSUB」のあとのコンマをとると、人間を動かすにはカーソル・キー、バリケードを落とすには空白キーでできるようになる。

実行すると、赤いインベーダーと青い人間が表示されて、敵のパワー（追いかける速度）を聞いてくる。パワーは1～100までで、数字が大きいほど、敵は速くなる。最初は10ぐらいがいいだろう。

数字を打ちこんで RETURN キーを押すと、ゲーム開始だ。画面には道ができて、左上に赤いインベーダー、右下に青い人間が表示される。インベーダーはすぐに追跡を始めるから、ジョイスティックを操作して人間を動かし、どんどん逃げよう。

追いつかれそうになったら、トリガボタンを押すと、◆の形をしたバリケードが出る。敵が◆を食べているあいだは時間をかせげるから、そのスキに逃げる。◆を落とすと10点加わり、インベーダーがそれを食べると10点減点されるから、結局、残った◆の数が点数になる。

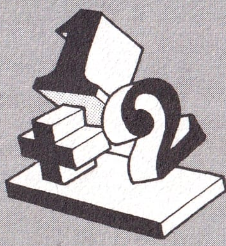
追いつかれたらゲーム終了。点数が表示される。

●プログラム

```
10 GOSUB 50
20 GOSUB 470
30 IF RND(1)>Z/100 THEN GOSUB 600
40 GOTO 20
50 '-------- ﾒﾆｭｰ ｶﾞﾒﾝ ﾋｮｳｼﾞ ------------
60 SCREEN 1,1:WIDTH 32:COLOR 15,1,1:CLS:KEY OFF
70 DD=RND(-TIME)
80 LOCATE 8,1:PRINT "ﾄｳﾎﾞｳ ｹﾞｰﾑ!!!!!!!"
90 GOSUB 930
100 PUT SPRITE 1,(10*8,4*8),5,1
110 LOCATE 13,5:PRINT "ｼﾞﾌﾞﾝ ﾃﾞｽ"
120 PUT SPRITE 0,(10*8,10*8),8,0
130 LOCATE 13,11:PRINT "ﾃｷ ﾃﾞｽ"
140 LOCATE 6,15:INPUT "ﾃｷ ﾉ ﾊﾟﾜｰ(1-100)";Z:Z=100-Z
150 '------- ｶﾍﾞ ﾄ ﾐﾁ ﾉ ﾋｮｳｼﾞ  ----------
160 SCREEN 1,0:WIDTH 32:COLOR 1,10,10
170 DEFINT A-Y
180 ZZ=.7
190 DIM AD(30,21)
200 FOR I=1 TO 21
210  FOR J=1 TO 30
220  LOCATE J,I:PRINT "ﾛ":AD(J,I)=9
230  NEXT J
240 NEXT I
```

```
250 FOR I=2 TO 20
260  M=I MOD 3
270  IF M=2 THEN M=1 ELSE M=3
280  FOR J=2 TO 29 STEP M
290   LOCATE J,I:PRINT " ":AD(J,I)=0
300  NEXT J
310 NEXT I
320 RESTORE 1020:GOSUB 930
330 '------ ショウトツ ノ ワリコミサキ ------------
340 ON SPRITE GOSUB 860
350 '------ トリガ゛ ボ゛タン ノ ワリコミサキ --------
360 ON STRIG GOSUB ,800
370 SPRITE ON
380 STRIG(1) ON
390 CX=2:CY=2:SX=29:SY=20
400 PUT SPRITE 0,(CX*8,CY*8),8,0
410 '------ ニンゲ゛ン ト インベ゛ーダ゛ ノ ヒョウジ゛----
420 PUT SPRITE 1,(SX*8,SY*8),5,1
430 DIM XV(8),YV(8)
440 XV(1)=0:XV(2)=0:XV(3)=1:XV(4)=0:XV(5)=0:XV(6)=0:XV(7)
    =-1:XV(8)=0
450 YV(1)=-1:YV(2)=0:YV(3)=0:YV(4)=0:YV(5)=1:YV(6)=0:YV
    (7)=0:YV(8)=0
460 RETURN
470 '------- ニンゲ゛ン ノ セイギ゛ョ ------------
480 SV=STICK(1)
490 IF SV=0 THEN RETURN
500 TX=SX+XV(SV):TY=SY+YV(SV)
510 IF TX<2 THEN TX=2
520 IF TX>29 THEN TX=29
530 IF TY<2 THEN TY=2
540 IF TY>20 THEN TY=20
550 IF AD(TX,TY)=1 THEN RETURN
560 IF AD(TX,TY)=9 THEN RETURN
570 SX=TX:SY=TY
580 PUT SPRITE 1,(SX*8,SY*8),5,1
590 RETURN
600 '------- インベ゛ーダ゛ ノ セイギ゛ョ ----------
610 TX=SX-CX:TY=SY-CY
620 WX=ABS(TX):WY=ABS(TY):WB=SGN(WX-WY)
630 IF WX>0 AND WY>0 THEN 680 ELSE IF WX>0 AND WY=0 THEN
    660
640 IF AD(CX,CY-1)=9 AND AD(CX,CY+1)=9 THEN IF TY>0 THEN
    IF AD(CX-1,CY+1)=9 THEN TX=1:TY=0 ELSE TX=-1:TY=0
    ELSE IF AD(CX-1,CY-1)=9 THEN TX=1:TY=0 ELSE TX=-1:
    TY=0 ELSE IF TY>0 THEN TX=0:TY=1 ELSE TX=0:TY=-1
650 GOTO 750
660 IF AD(CX-1,CY)=9 AND AD(CX+1,CY)=9 THEN IF TX>0 THEN
    IF AD(CX+1,CY+1)=9 THEN TX=0:TY=-1 ELSE TX=0:TY=1
    ELSE IF AD(CX-1,CY+1)=9 THEN TX=0:TY=-1 ELSE TX=0:
    1 ELSE IF TX>0 THEN TX=1:TY=0 ELSE TX=-1:TY=0
```

```
670 GOTO 750
680 IF TX>0 AND TY>0 THEN IF WB>0 THEN IF AD(CX+1,CY)=9
    THEN TX=0:TY=1 ELSE TX=1:TY=0 ELSE IF AD(CX,CY+1)=9
    THEN TX=1:TY=0 ELSE TX=0:TY=1 ELSE 700
690 GOTO 750
700 IF TX>0 AND TY<0 THEN IF WB>0 THEN IF AD(CX+1,CY)=9
    THEN TX=0:TY=-1 ELSE TX=1:TY=0 ELSE IF AD(CX,CY-1)=9
    THEN TX=1:TY=0 ELSE TX=0:TY=-1 ELSE 720
710 GOTO 750
720 IF TX<0 AND TY>0 THEN IF WB>0 THEN IF AD(CX-1,CY)=9
    THEN TX=0:TY=1 ELSE TX=-1:TY=0 ELSE IF AD(CX,CY+1)=9
    THEN TX=-1:TY=0 ELSE TX=0:TY=1 ELSE 740
730 GOTO 750
740 IF WB>0 THEN IF AD(CX-1,CY)=9 THEN TX=0:TY=-1 ELSE
    TX=-1:TY=0 ELSE IF AD(CX,CY-1)=9 THEN TX=-1:TY=0 ELSE
    TX=0:TY=-1
750 TX=CX+TX:TY=CY+TY
760 IF AD(TX,TY)=1 THEN IF RND(1)<ZZ THEN PUT SPRITE 0,
    (CX*8,CY*8),6,0:RETURN ELSE AD(TX,TY)=0:LOCATE TX,TY
    :PRINT " ";:PLAY "l64o2f":SC=SC-10:LOCATE 0,23:
    PRINT USING "○○○ SCORE ##### ○○○";SC;
770 CX=TX:CY=TY
780 PUT SPRITE 0,(CX*8,CY*8),8,0
790 RETURN
800 '------- ﾄﾘｶﾞ ﾎﾞﾀﾝ ﾉ ﾜﾘｺﾐｼｮﾘ --------
810 IF AD(SX,SY)=1 THEN RETURN
820 LOCATE SX,SY:PRINT "◆";:PLAY "L6407D"
830 AD(SX,SY)=1
840 SC=SC+10:LOCATE 0,23:PRINT USING "●●● SCORE ##### ●●●
    ";SC;
850 RETURN 20
860 '------- ｼｮｳﾄﾂ ﾉ ﾜﾘｺﾐｼｮﾘ ------------
870 STRIG(1) OFF:COLOR 1,14,6:PLAY "L4004BAGFEDCD"
880 FOR I=1 TO 24:FOR J=1 TO 10:NEXT J:PRINT:NEXT I
890 COLOR 15,1,7
900 LOCATE 8,10:PRINT "G A M E  O V E R"
910 LOCATE 6,13:PRINT USING "●●● SCORE ##### ●●●";SC;
920 END
930 '------- ﾆﾝｹﾞﾝ ﾄ ｲﾝﾍﾞｰﾀﾞ ﾉ ｻｸｾｲ -----
940  FOR I=0 TO 1
950   S$=""
960   FOR J=1 TO 8
970    READ A$:S$=S$+CHR$(VAL("&H"+A$))
980   NEXT J
990   SPRITE$(I)=S$
1000  NEXT I
1010 '----- ﾆﾝｹﾞﾝ ﾄ ｲﾝﾍﾞｰﾀﾞ ﾉ ﾃﾞｰﾀ -----
1020  DATA 18,3C,DB,3C,18,3C,42,81,11,3A,14,38,50,98,24,48
1030 RETURN
```



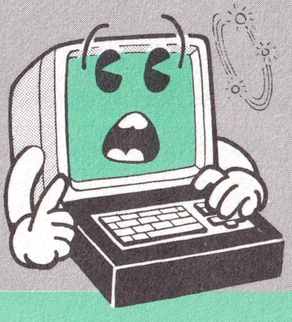
·PART·
6

パソコンでできる

暗算練習



パソコンの主要な仕事のひとつに演算があります。しかもきわめて高速に処理するし、正確です。ここでは、この演算についてお話しします。とはいえ、ただ計算させるだけでは電卓と同じですから、パソコンが、人間の出した答えの正誤を判断するようなこともやらせてみます。この章で学ぶ命令は、INPUT、IF～THEN～ELSE、BEEP、READ～DATA、ON～GOTO、RESTORE、です。



LESSON ①

パソコンは便利な計算機

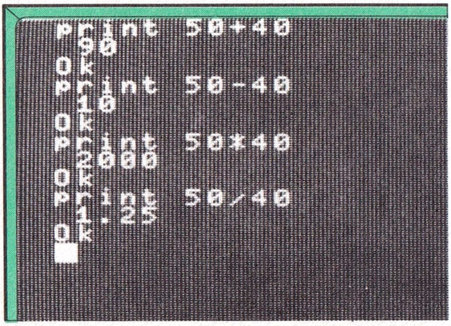
まず電卓的な使い方から

直接モードと間接モードの違いについては、すでに説明しました（54ページ参照）。パソコンが四則演算をどのように行なうか、直接モードを使って調べてみましょう。次のようにキーインしてください。

```
①PRINT␣50+40 RETURN
②PRINT␣50−40 RETURN
③PRINT␣50*40 RETURN
④PRINT␣50／40 RETURN
```

実行画面と合わせて見れば、どの記号が、一般の数学での、どの演算記号になるかがわかるでしょう。これらの記号を算術演算子といいます。

乗算と除算の記号は数学の記号とは形は違いますが、使い方は同じですからすぐに慣れるでしょう。正の数の頭につくプラス符号(＋)は省略できます。負の数のマイナス符号(−)は省略できません。また、数値になんの符号もついていなければ、正の数とみなします。



ここまで説明すればわかるように、直接モードによる演算で使われる表示命令 **PRINT**（プリント）は、たんに「画面に表示せよ」ではなく、「表示命令以下の演算を行ない、その結果を表示せよ」という命令になります。

MSXパソコンで扱える数値は、±9.9999999999999×10^{62}の範囲です。これを超えると「Overflow（オーバフロウ）」というエラーメッセージが出ます。また、0による除算を実行した場合は、「Division（ディビジョン） by（バイ） zero（ゼロ）」というエラーメッセージが出ます。どちらの場合も処理が中断し、入力待ちの状態になります。

◉算術演算子の種類

種　類	呼　称	数学記号
＋	プラス	＋
−	マイナス	−
＊	アスタリスク	×
／	スラッシュ	÷

パソコンでこんな計算もできる

✣累乗計算(例 1)

四則演算以外の演算を行なう演算子もあります。2^3、6^4 など累乗(るいじょう)の計算には「∧」(アップアロー)記号を使い、「2∧3、6∧4」などとして計算します。

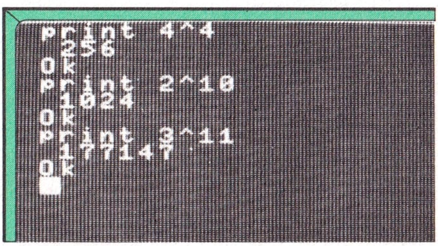
✣整数の除算(例 2)

除算を整数で行なうときには、「／」(スラッシュ)のかわりに¥記号を使用します。除算の数字の小数点以下を切りすてて演算すると同時に、商の小数点以下が切りすてられます。

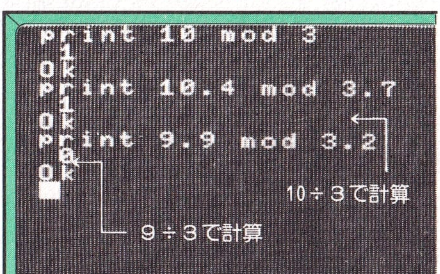
✣除算の余りだけを計算する(例 3)

除算の余りだけを計算することもできます。スラッシュのかわりにMOD(モッド)記号を用いると余りが出ますが、この場合は除算の数値の小数点以下を切りすててから演算を実行します。したがって、余りはつねに整数になります。

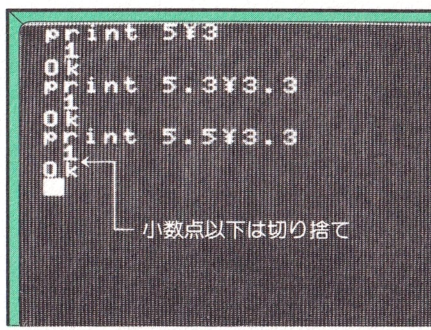
例 1 ∧を使って、4^4、2^{10}、3^{11}、を計算する

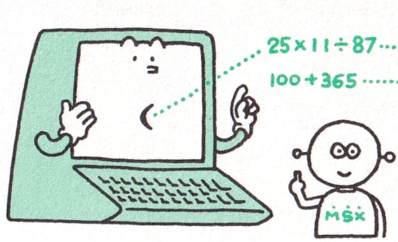


例 3 MOD を使って、10÷3、10.4÷3.7、9.9÷3.2の余りを求める



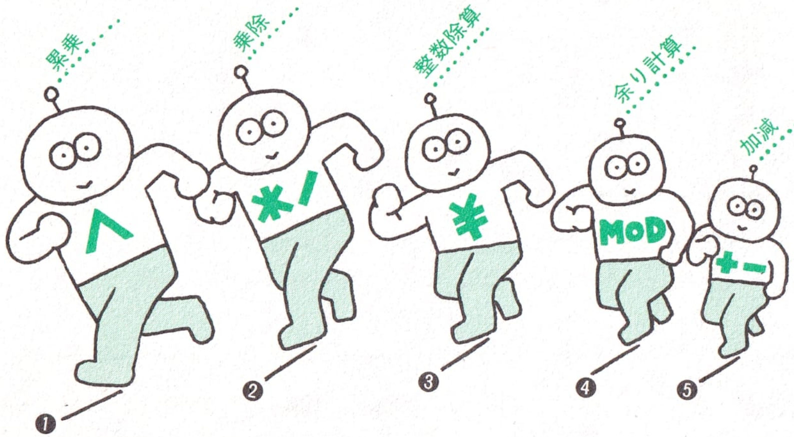
例 2 ¥を使って、5÷3、5.3÷3.3、5.5÷3.3を計算する。除算の数字、商ともに、小数点以下が切りすてられる





計算の順序を調べてみると

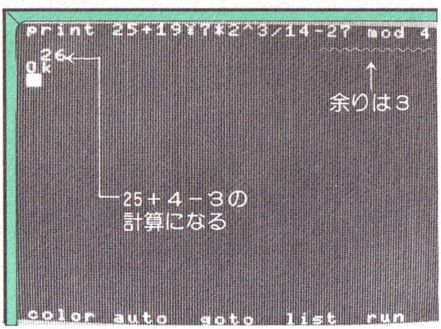
算術演算子は、例で示した7種類ですべてです。これらを同じ命令文の中でいっしょに用いた場合には、次のように、一定の優先順位に従って実行されます。



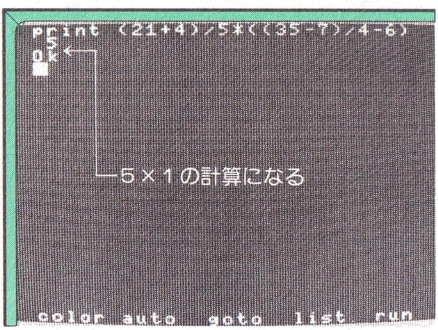
▲算術演算子の実行順序

また、カッコをつけた場合には、数学の演算と同様、カッコ内が優先して実行されます。カッコを幾重にも使用する場合も数学と同様ですが、(、{、[などの区別はなく、すべて (を使います。

例 PRINT␣25＋19¥7＊2∧3／14－27 MOD 4の命令文を実行する



例 PRINT␣(21＋4)／5＊((35－7)／4－6) の命令文を実行する



LESSON ②

いろいろなデータでどんどん演算

直接モードからプログラムへ

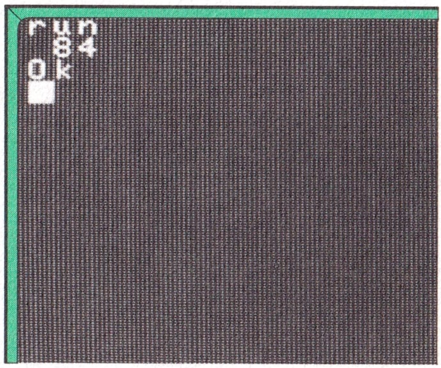
算術演算子をマスターしたら、この章の課題に順を追ってとりくむことにします。この章の最終的な課題は、「パソコンが出した問題に人間が答え、その答えをパソコンが判断して正誤の判定をくだすプログラムをつくる」ことです。そのための第１ステップとして、間接モードでの演算プログラムを作成します。

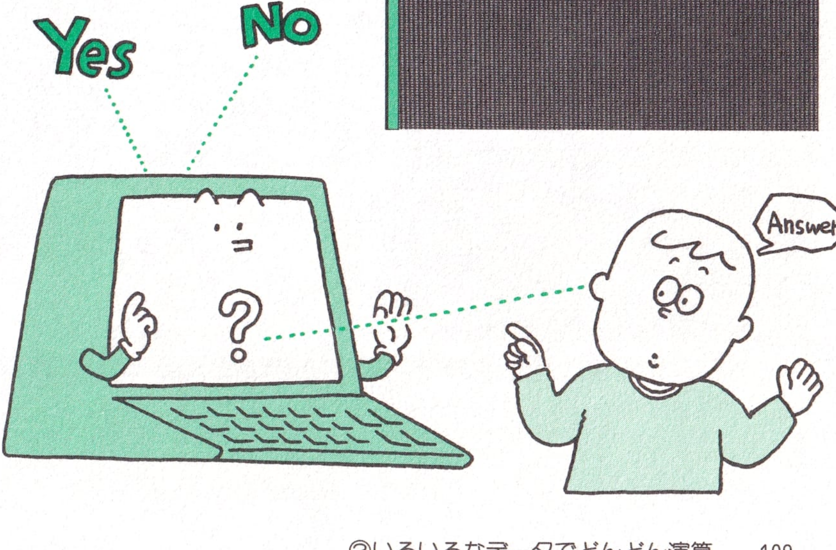
例として「64＋20」の演算式を使います。間接モードで「64＋20 の演算をして、その結果を表示する」という課題を行なうためには、直接モードで考えてから命令文の先頭に行番号をつけ、最後に終止命令 **END** の行を追加します。

●プログラム

```
10 PRINT 64+20
20 END
```

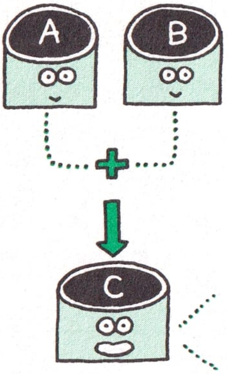
→ 実行





問題と答えを表示するプログラムをつくる

しかし、実行結果だけでは、「84」という値がどんな演算の結果なのかよくわかりません。そこで、演算式と結果の両方を表示するようにします。それには代入命令をうまく使うくふうが必要です。AとB、その結果にCという変数を使うとすると、手順は次のようになります。



①Aに64を代入する。
②Bに20を代入する。
③A+Bを演算して、その結果をCに代入する。
④式と演算結果を表示する。

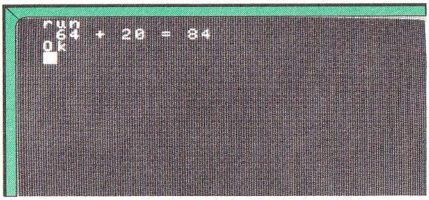
プログラムを作成する際には、手順④がたいせつです。表示命令**PRINT**（プリント）のあと、変数はそのまま書けば代入された値が表示されるのに対して、+や−の記号は文字として扱われるので、ダブルコーテーションをつけなくてはなりません。

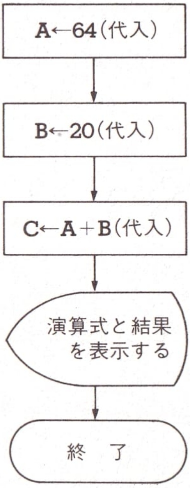
●プログラム

```
10 A=64
20 B=20
30 C=A+B
40 PRINT A;"+";B;"=";C
50 END
```

（ダブルコーテーション："+" と "=" の引用符を指す注記）

実行





練習問題 6

次の演算式を行ない、演算式と結果を表示するプログラムをつくってみよう。

①56+27×2
②64÷2+32−51

値を変えて計算するにはINPUT（インプット）

数学や物理などでは、同じ公式を使ってさまざまな演算を行なうことがよくあります。ごく単純な形として前のプログラムで使った「A＋B」を公式として考えてみましょう。これまでの方法では、AとBをさまざまな値にして演算する場合には、そのたびに同じプログラムを、AとBにさまざまな値を入れてつくらなくてはならず、たいへんにめんどうです。

そこで、AとBの値をキーボードから入力するだけで、演算を実行する方法を知っておくと便利です。前出のプログラムを、もう一度入力してください。このプログラムの行10と行20を、次のように修正します。

```
10 INPUT A
20 INPUT B
```

修正が終わったら、正しく修正できたかどうかリストをとって確認してください。正しく修正できていれば、一度画面を消去して実行します。すると「？」のマークが表示されるので、適当な数字をキーボードから打ちこみ、入力終了キー RETURN を押します。これで変数Aに値が入力されます。再び？マークが出てきたら、同じ操作（値は適当に）をくり返します。これで、Bに値が代入されます。この操作が終了すると、演算式と結果が表示されます。

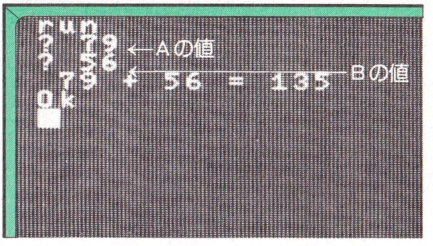
たとえば、Aを79、Bを56とキーインすると、下のように画面に表示されます。これで再び実行すれば、また、？マークが出てきます。こんどは違う値を入力すれば、その値で演算することができます。こうして、次々に演算処理ができるわけです。

ここで注意したいのは、？マークに続いて適当な値をキーインしたあとは、必ず、入力終了キー RETURN を押すことです。さもないと、その値は変数に代入（記憶）されません。

●プログラム

```
10 INPUT A
20 INPUT B
30 C=A+B
40 PRINT A;"+";B;"=";C
50 END
```

→ 実行



複数のデータ入力には2とおりの方法がある

入力命令 **INPUT**（インプット）は、パソコンに仕事をさせるときによく使うので、もう少し説明を続けます。完全にマスターしてください。演算式をもう少し複雑にして考えてみましょう。

「変数A、B、C、Dの値を入力して、A＊B+C／Dの演算をし、その演算式と結果を表示する」という課題を例にすると、プログラムは下のようになります。

行10～40は、1行に縮めることができます。変数をコンマ（，）でつなげば、**INPUT**（インプット）は一度使うだけですむのです。

```
10 INPUT A, B, C, D
```

左のように修正し、行20～40を消去して、実行します。すると、前と同様に？マークが出てきます。ここで、データの入力は、2とおりの方法があります。どちらの方法を使ってもよいでしょう。

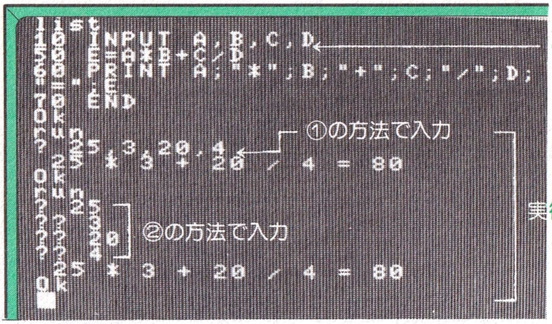
①A、B、C、Dのデータをコンマで区切って入力し、最後に入力終了キー RETURN を押す。

②A、B、C、Dのデータをひとつずつ入力終了キーを押しながら入力する。

●プログラム

```
10 INPUT A
20 INPUT B
30 INPUT C
40 INPUT D
50 E=A*B+C/D
60 PRINT A;"*";B;"+";C;"/";D;"=";E
70 END
```

修正して実行



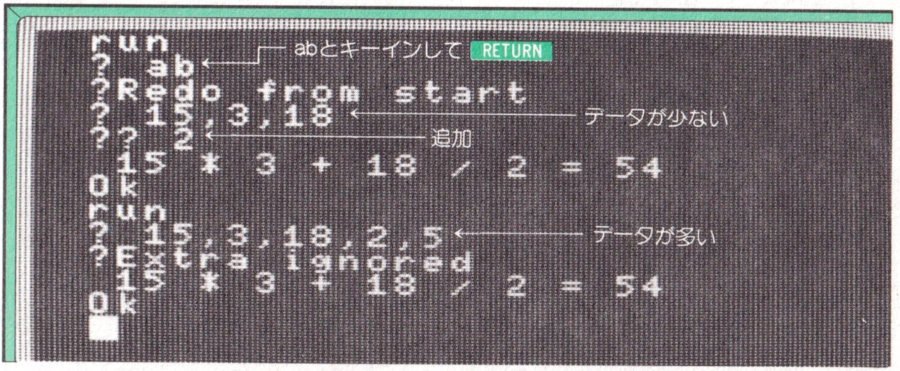
入力は変数の型と数に注意して

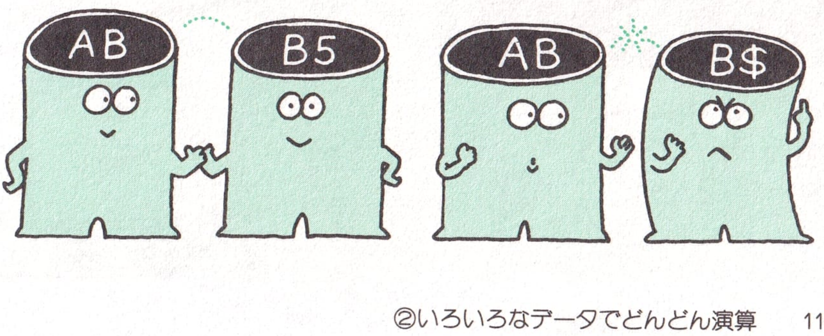
INPUT を使って、データを入力する場合には、変数の型（数値型と文字型の別）や数に注意して、入力すべき変数の型、数と一致させるようにします。

数値型変数なのに文字を入力したり、あるいは入力したデータの数が変数の数より少なかったりすると、？マークのあとに「Redo from start」という文字が表示されます。「最初からデータを入れなおしてください」という意味のエラーメッセージです。

このエラーメッセージが出た場合は、そのすぐ下に？マークが出て、入力待ちの状態になります。そのあとで正しいデータを入力すれば、それ以降は正しく実行されます。

また、変数の数より入力したデータの数が多い場合は、「？Extra ignored」（データがよぶん）というエラーメッセージが出て、よぶんなデータが無視され、次の行が実行されます。逆に、データが不足したときには、「??」という表示が出て、不足しているデータが入力されるのを待ちます。





入力ミス防止はプロンプト文で

入力ミスを防ぐのに、有効な方法があります。？マークがなんのデータを聞いているのか表示するのです。下の行10と行20のように、ダブルコーテーションではさんだ文字を **INPUT** の後ろに入れます。こうすると、なんのデータを入れたらよいのかがひと目でわかります。

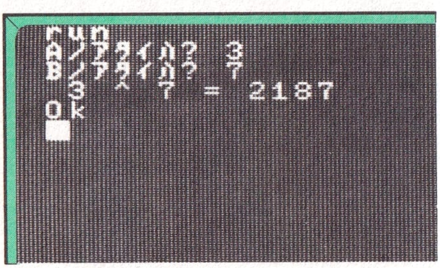
このプログラムを実行すると、？マークの前に入力データを指定する文が表示されます。これならば、プログラムをつくった人でなくとも、まちがいなく入力できるでしょう。

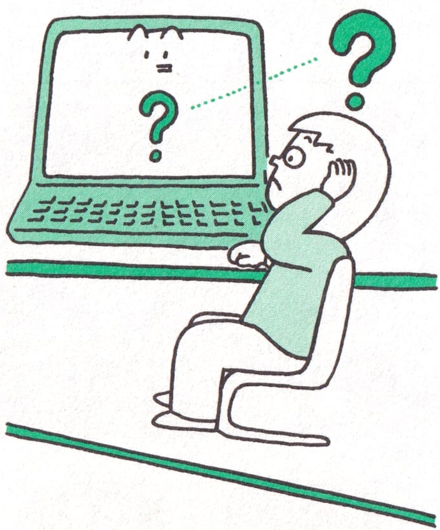
INPUT のあとにダブルコーテーションではさまれた文を「プロンプト文」といいます。プロンプト文があると、**INPUT** は、表示命令 **PRINT** と同じような働きもするのです。

●プログラム

```
10 INPUT "Aノアタイハ";A
20 INPUT "Bノアタイハ";B
30 C=A^B
40 PRINT A;"^";B;"=";C
50 END
```

実行







入力命令の使用にはこれだけの注意を

入力命令 **INPUT**（インプット）の使い方に慣れてきたでしょうか。ここで、入力命令 **INPUT**（インプット）を使用したプログラムと、その実行結果をあげておきましょう。プログラムは複利計算をするためのもので、例として元金１万円、年利率６％で３年間預けたときの元利合計を算出します。入力命令 **INPUT**（インプット）を使うプログラムを作成するときには、次のような点に注意します。

①**INPUT**（インプット）のあとの変数は、入力すべきデータが数値のときには数値型変数、文字のときには文字型変数（$つき）にする。

②プロンプト文はダブルコーテーション（" "）ではさみ、変数とのあいだにはセミコロン（；）をつける。

③**INPUT**（インプット）のあとに複数の変数を書く場合には、変数ごとにコンマ（，）で区切る。

また、データを入力するときのポイントは、次のとおりです。

●プログラム：預金の複利計算

```
10 INPUT "ｶﾞﾝｷﾝﾊ";G
20 INPUT "ﾘﾘﾂﾊ";R
30 INPUT "ﾈﾝｽｳﾊ";N
40 K=G*(1+R/100)^N
50 PRINT "ｶﾞﾝﾘｺﾞｳｹｲ";K
60 END
```

実行

```
ｶﾞﾝｷﾝﾊ? 10000
ﾘﾘﾂﾊ? 6
ﾈﾝｽｳﾊ? 3
ｶﾞﾝﾘｺﾞｳｹｲ 11910.16
Ok
```

①データをキーインしたら、必ず RETURN キーを押す。

②変数の型（数値型変数と文字型変数の別）に合わせたデータを入力する。

③変数の数に合った個数だけ、データを入力する。

基本形

INPUT（インプット） **"プロンプト文"; 変数, 変数……**

キーボードから入力したデータを、指定した変数に代入する入力命令

練習問題７

入力命令を使って、次のようなプログラムを作成してみよう。

①打数と安打数を入力すると打率が出るプログラム。

②利率と元金を入力すると、１年ごとの元利合計が10年めまで出るプログラム。

【ヒント】 ②はくり返し命令 **FOR～NEXT**（フォー・ネクスト）を使います。

LESSON ③

パソコンがする採点と判断

パソコンにも問題が出せる

入力命令の説明が終わったところで、本章の本題である「コンピュータが問題を出題し、人間が答えを入力する」という課題に入ることにします。

問題を出題する手順は、次のようになります。問題は「64＋20」とします。

①変数 A に 64 を代入する。

②変数 B に 20 を代入する。

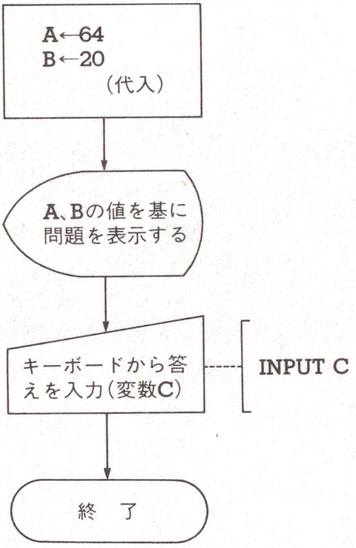
③A、B の値をもとに問題を表示する。

このプログラムは、すぐにできるでしょう。次のようになります。

```
10 A=64
20 B=20
30 PRINT A;"+";B;"="
40 END
```

ここに、人間が答えを入力するという手順を加えます。手順の続きでいえば、次の④になります。これで、いちおうの形としては、「パソコンが問題を出し、人間が答える」プログラムができました。

④A＋B の答えを人間が入力する。



●プログラム

```
10 A=64
20 B=20
30 PRINT A;"+";B;"="
40 INPUT C
50 END
```

実行

```
run
 64 + 20 =
? 84
Ok
```

パソコンは、判断命令で考える

前項のプログラムを、パソコンの演算結果（正解）と人間の答えとを比べ、正誤の判別を表示するように修正します。正誤判別の手順は、次のとおりです。

①コンピュータの演算結果（A+B）と人間の答え（C）をつきあわせる。

②もし、C=A+Bならば、正解であることを表示する。

③もし、C=A+Bでなければ、誤りであることを表示する。

パソコンが電卓とまったく違う機械であることを教えてくれるのが、本項で説明する判断命令 **IF～THEN～ELSE**（イフ・ゼン・エルス）です。**IF～THEN**（イフ・ゼン）という形については、すでに説明しました（89ページ参照）。この命令によって、パソコン自身がさまざまな判断を行ないます。

基本形のうち、条件式とは「もしC=A+Bであれば」あるいは「もしC=A+Bでなければ」の「C=A+B」のことをいいます。判断命令では、条件式の条件に適合していれば **THEN**（ゼン）のあとの命令文を実行し、適合していなければ **ELSE**（エルス）のあとの命令文を実行します。

条件判断を行なうプログラムを作成するには、①条件式　②条件に適合した場合の処理　③条件に適合しない場合の処理、の3つを明確にして、次のようなプログラムをつくります。

IF　①　THEN　②　ELSE　③

上の処理のフローチャートと、**ELSE**（エルス）以下を書かない場合のフローチャートをあげておきます(次ページ)。

条件式の中のイコール(=)は、代入命令の場合のイコールと違い、数学と同じく「等しい」ことを意味します。条件をあらわす記号は、基本的なものとして、「=」のほかに「>」、「<」があり、さらにこれらを組み合わせたものとして「<

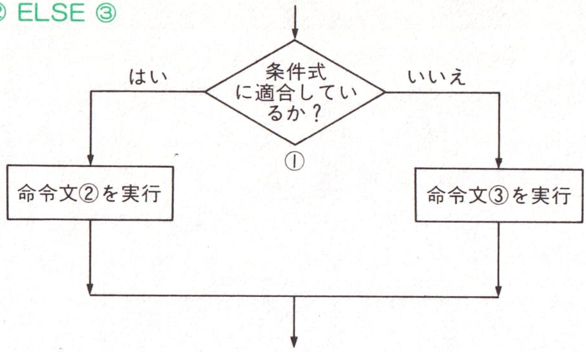
基本形

IF（イフ）␣**条件式**␣**THEN**（ゼン）␣**命令文**␣**ELSE**（エルス）␣**命令文**

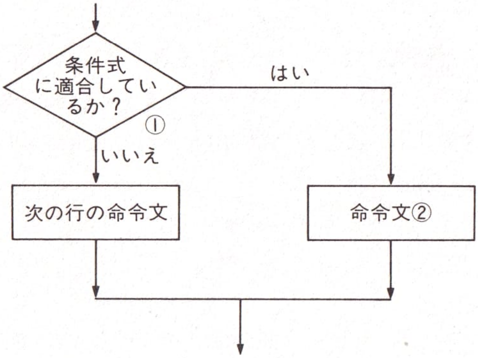
条件式の条件を判断し、実行する命令文を決定する判断命令

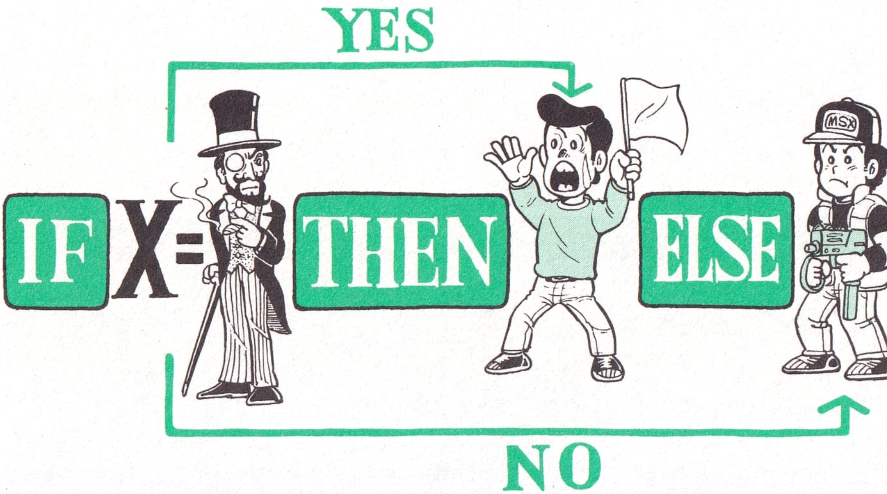
＞」、「＞＜」、「＜＝（＝＜）」、「＞＝（＝＞）」があります。これらを「関係演算子」と呼びます。

IF ① THEN ② ELSE ③



IF ① THEN ②





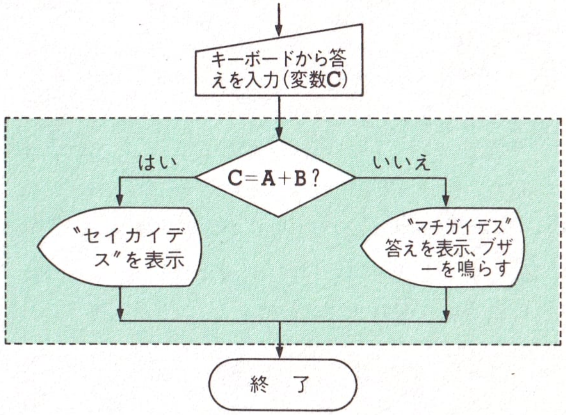
正否の判断をするプログラムは

前のプログラムに手を加えて、行40で入力した人間の答えが正解かどうかの判断を、次のようにします。

①正解ならば「セイカイデス」と表示する。

②誤りならば「マチガイデス　コタエハ」と表示して正解を示し、ブザーを鳴らす。

フローチャートで示した部分で、この①、②を処理します。

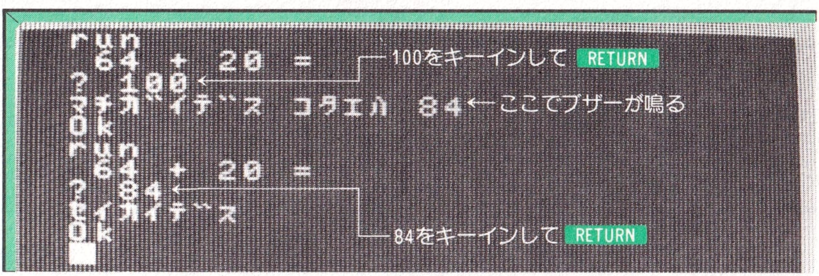


行50は、マルチステートメント、つまり多重命令になっています。ELSE(エルス)のあとに、コロン(：)でつながれた2つの命令が、入っているということです。THEN(ゼン)のあとも、マルチステートメントにすることが可能です。

●プログラム：正否の判断をする

```
10 A=64
20 B=20
30 PRINT A;"+";B;"="
40 INPUT C
50 IF C=A+B THEN PRINT "ｾｲｶｲﾃﾞｽ" ELSE PRINT "ﾏﾁ
   ｶﾞｲﾃﾞｽ ｺﾀｴﾊ";A+B:BEEP
60 END
```

実行



判断命令文のいろいろな書き方

判断命令文の書き方は、必ずしも先にあげたような形でなくてもかまいません。たとえば、判断命令のうち **ELSE** を使わず、前出プログラムの行50以下を、次のように変えることもできます。

```
50 IF C=A+B THEN PRINT "セイカイ　デス":GOTO 70
60 PRINT "マチガイ　デス　コタエハ";A+B:BEEP
70 END
```

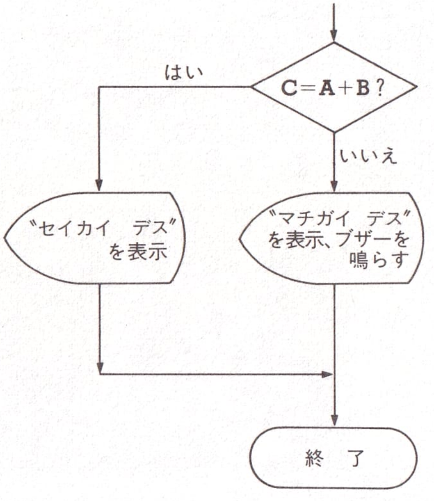
フローチャートを見ると、正解ではない場合の処理の流れが違います。そのまま下の処理（終了）を実行するわけです。正解の場合は、「セイカイ　デス」を表示したあと、「GOTO　70」によって、終止命令文に飛びこします。この処理をしておかないと、「セイカイ　デス」の表示のあとに、「マチガイ　デス」という表示が出て、おかしなことになります。

判断命令の **THEN** と **ELSE** のあとには、行番号を書くこともできます。

```
IF G=F THEN 70 ELSE 80
```

これにより、条件に適合する場合は行70へ飛び、適合しない場合は行80へ飛び、それ以降の命令を実行します。この場合、**THEN** と **ELSE** 以降がマルチステートメントのときは、行番号の前に飛びこし命令 **GOTO** をつけます。

前出プログラムの行50にある **BEEP** は、約0.04秒間、ブザーを鳴らすブザー命令です。



基本形

BEEP

ブザーを鳴らすブザー命令

LESSON ④

問題が次々に出てくるプログラム

データの読みこみには READ～DATA

あらかじめコンピュータの中に多くの演算問題を入れておいて、それを次々に表示する形にすると、パソコンはかっこうの暗算練習機になります。それには、演算の変数にさまざまな数値を入れます。

これまで、変数に数値を与える方法として、代入命令 **LET** と入力命令 **INPUT** を説明しました。ここでは、読みこみ命令 **READ～DATA** について解説します。

READ と **DATA** は、本来は2つの命令です。狭い意味でいえば、**READ** が読みこみ命令であり、**DATA**は「数値や文字定数を **READ** で読みこまれるデータとして定義する」命令です。ただし、実際には2つがつねに組み合わせて使われるので、ここでは1つの命令として扱うことにします。

読みこみ命令は、**READ** 文に変数、**DATA** 文に定数を書きます。変数や定数がいくつもある場合には、コンマ(，)で区切ります。もっとも初歩的なプログラムをつくり、その働きを見てみましょう。

行30の **DATA** 文の「64」と「20」を行10の **READ** 文が読みこみ、変数Aには64が、変数Bには20が代入されます。この数値に基づいて、行20が実行されます。

●プログラム

```
10 READ A,B
20 PRINT A;"+";B;"=";A+B
30 DATA 64,20
40 END
```

実行

```
run
 64 + 20 = 84
Ok
```

基本形

READ␣変数，変数，変数……
～
DATA␣定数，定数，定数……

プログラムの中から DATA 文を探し出して、そこから値を読みこみ、変数に割りあてる読みこみ命令

読みこみ命令はREADとDATAの対応に注意

読みこみ命令を使って、もう少し複雑な仕事をさせてみましょう。

次のプログラムの行50の**DATA**文は、行10の「READ　A, B」で最初の2数値（72と32）を、行30の「READ　C, D」で次の2数値（63と3）を読みこんでいます。このように、**READ**文を2つ書いても、**DATA**文の定数は順番に読みこまれます。**READ**文を1行にしても結果は同じです。その場合は、行10と行30とをまとめて次のように書きます。

```
10 READ A,B,C,D
```

このように、**READ**文の変数の順番（左から）と、**DATA**文の定数の順番とは対応しています。もし、**READ**文の変数の数より**DATA**文の定数の数が少ない場合は、「Out of data」（データ不足）というエラーメッセージが出ます。逆に、**DATA**文の定数の数のほうが変数より多い場合は、エラーメッセージは出ませんが、よぶんな定数は読みこまれません。

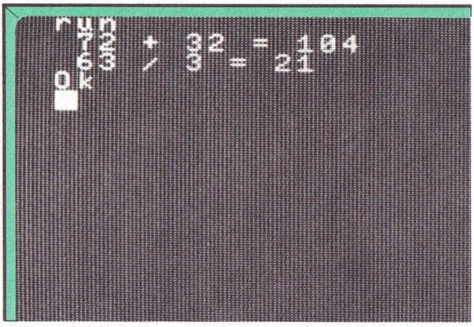
また、**READ**文と**DATA**文の変数の型（数値型と文字型の別）は、互いに一致させなくてはなりません。

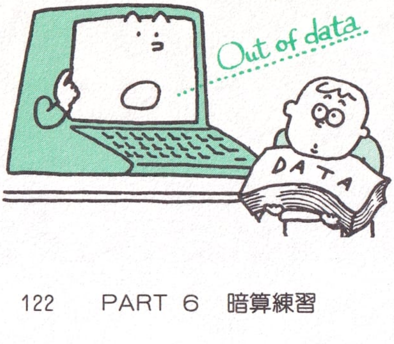
DATA文は、プログラム中のどこに書いてもかまいません。**READ**文が**DATA**文を探して読みこみます。終止命令**END**のある行の下に書いてもよいし、**READ**文の上に書いても結果は同じです。

●プログラム

```
10 READ A,B
20 PRINT A;"+";B;"=";A+B
30 READ C,D
40 PRINT C;"/";D;"=";C/D
50 DATA 72,32,63,3
60 END
```

実行

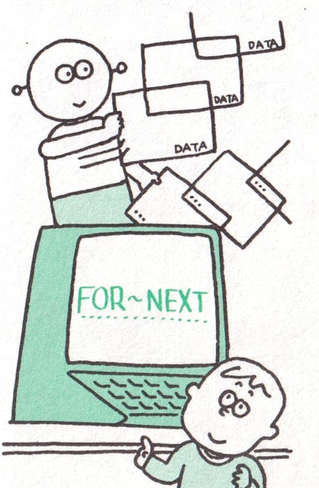


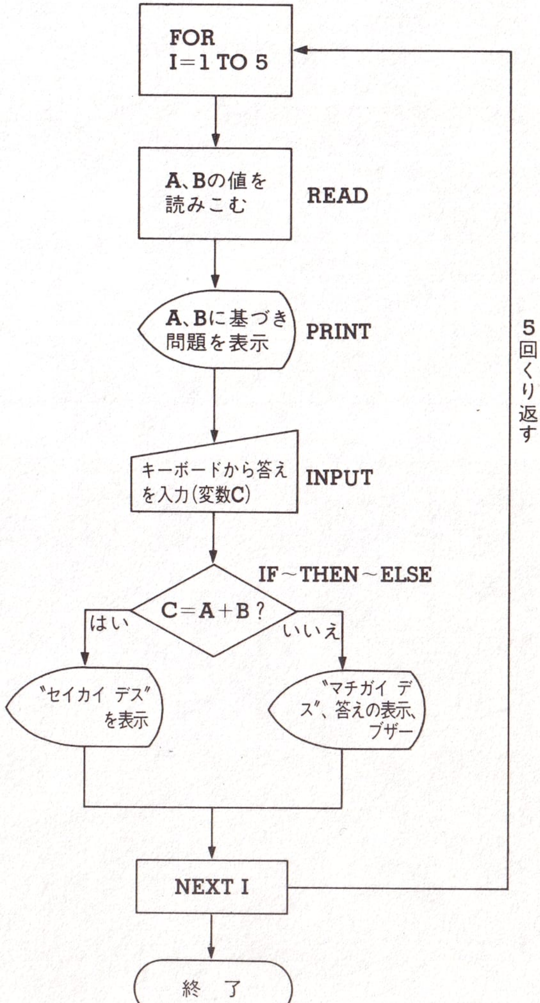


便利な「読みこみ」＋「くり返し」

読みこませるデータが多い場合には、くり返し命令**FOR～NEXT**（フォー・ネクスト）を使って読みこみをくり返します。その際、**DATA**（データ）文のデータの数と、読みこんだデータの数を同じにするよう注意します。たとえば、変数がAとBの2つで、読みこみ命令を5回くり返す場合は、データの数は10個になります。

ここで、パソコンが「A＋Bは？」という問題を出し、人間が答えを入力して、その正誤を判定するプログラムをつくってみましょう。**DATA**（データ）文は処理の流れには関係しないので、フローチャートには書かないのがふつうです。フローチャートには、それぞれの処理に必要な命令（**READ**（リード）など）を記入しておくと、プログラムをつくるのに便利です。





これを実行すると、5種類の問題が次々に表示されます。人間がそのたびに答えを入力すると、パソコンが正誤を判断し、知らせてくれます。

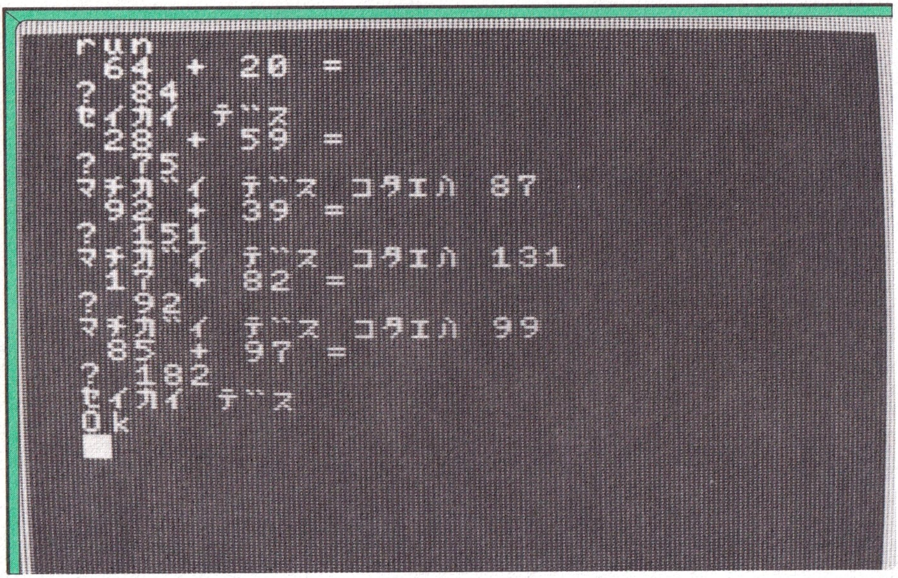
また、DATA（データ）文は、次のように1行にすることもできます。

```
80  DATA  64, 20, 28, 59, 92, 39, 17, 82, 85, 97
```

●プログラム

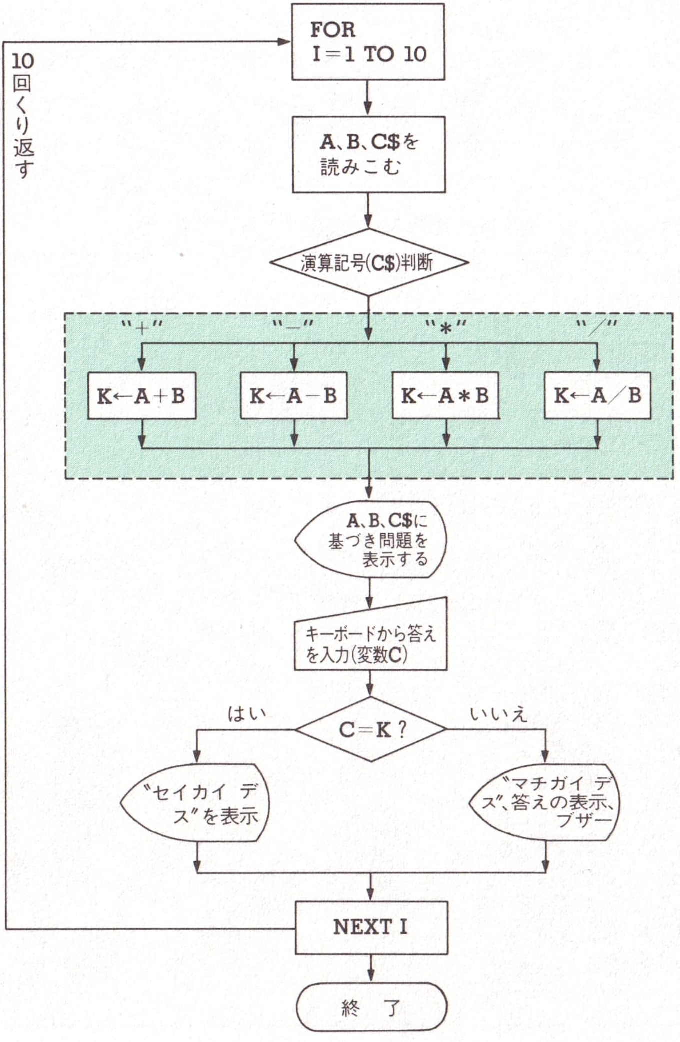
```
10 FOR I=1 TO 5
20     READ A,B
30     PRINT A;"+";B;"="
40     INPUT C
50     IF C=A+B THEN PRINT "ｾｲｶｲ ﾃﾞｽ" ELSE PRINT
       "ﾏﾁｶﾞｲ ﾃﾞｽ ｺﾀｴﾊ";A+B:BEEP
60 NEXT I
70 END
80 DATA 64,20
90 DATA 28,59
100 DATA 92,39
110 DATA 17,82
120 DATA 85,97
```

実行



加減乗除の問題を出す形に

加算の問題だけでなく、減算や乗算、除算などの問題も出題されるようにするには、**DATA**(データ) 文の中に、それらの問題を加えればよいのです。フローチャートの中で点線で囲んだのは、演算の種類（条件）によって実行の内容を変える処理を行なう部分です。条件を判断するためには判断命令 **IF〜THEN**(イフ・ゼン) を使います。



これで暗算練習のプログラム

フローチャートに従って、プログラムをつくれば、これで暗算練習のプログラムはできあがります。プログラムをつくるときに注意したいのは、**READ** 文の中の変数に、数値と文字があることです。文字型変数はダラー記号（$）のついた変数を使います。この場合には、**DATA** 文中のデータも、文字型変数に合った順番のところには文字がきます。くり返しますが、**READ** 文の中の変数と **DATA** 文の定数の型（数値型と文字型の別）は一致させなくてはなりません。一致していないと、「Syntax error」が表示され、実行が終了します。

プログラムでは、演算の種類に関係なく、答えをすべて「K」というひとつの変数に代入していることに注目してください（行 30～60 の **THEN** の後ろ）。こうすることによって、正誤の判断をする命令文は行 90 だけですむわけです。

●プログラム

```
10 FOR I=1 TO 10
20    READ A,B,C$
30    IF C$="-" THEN K=A-B:GOTO 70
40    IF C$="*" THEN K=A*B:GOTO 70
50    IF C$="/" THEN K=A/B:GOTO 70
60    IF C$="+" THEN K=A+B
70    PRINT A;C$;B;"="
80    INPUT C
90    IF K=C THEN PRINT "ｾｲｶｲ ﾃﾞｽ" ELSE PRINT
      "ﾏﾁｶﾞｲ ﾃﾞｽ ｺﾀｴﾊ";K:BEEP
100 NEXT I
110 END
120 DATA 10,10,+
130 DATA 20,30,*
140 DATA 10,2,/
150 DATA 10,3,-
160 DATA 25,36,+
170 DATA 50,32,-
180 DATA 18,3,/
190 DATA 76,2,*
200 DATA 64,44,+
210 DATA 54,2,-
```

正解数をかぞえるには「S＝S＋1」

入力した答えの正解率を計算し、表示するプログラムをつくります。

正解率を表示するためには、

①正解の数をかぞえる。

②正解率を計算する。

③正解率を表示する。

という3つの手順が必要です。②と③は簡単でしょう。正解数をS、正解率をRとすると、それぞれ次のような命令文になります。問題数は10とします。

```
②――R=S／10＊100
③――PRINT　R;"%　デス"
```

この2つは1行にできます。

```
PRINT　S／10＊100;"%　デス"
```

さて、問題の①です。Sをどう数えるか。Sの値がどう変化するか考えてみましょう。Sは正解ごとに1ずつ増えていきます。

1問正解　　S=1

2問正解　　S=2（1+1）

3問正解　　S=3（2+1）

　⋮　　　　　⋮

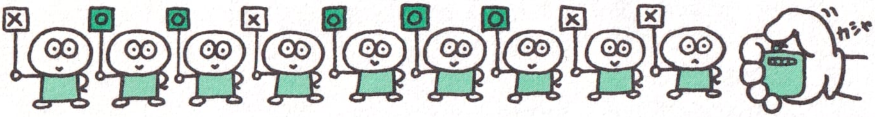
つまり、Sの値は、1問正解するごとに、すぐ前までの正解数に1を加算する形で増えていきます。したがって、代入文は、次のような形になります。

```
S=S+1
↑ ↑
│ └────すぐ前までの正解数
└────現在の正解数
```

プログラムは、1問正解するごとに、この代入文が実行されるよう作成します。もうひとつ、プログラムを実行した最初は、Sの値が0でなければなりません。そうでないと、正解率が正しく計算されません。ただし、「S=0」という代入文は省略できます。数値型変数は、実行して初めて処理されるときには、自動的に初期値が0となるからです。

「S=S+1」の位置に注意してプログラムに

プログラムをつくるときには、「S=S+1」の位置に注意しましょう。位置によっては、まちがいの数までかぞえたり、たんに問題数をかぞえるだけに終わったりします。一般的にいって、正解数をかぞえるのは、正解の表示が出た直後にすれば、まずまちがいありません。逆に、まちがいの数をかぞえるには、まちがいの表示が出たすぐあとに置けばよいのです。したがって、下のプログラムのような位置に「S=S+1」がきています。



●プログラム

```
10 S=0
20 FOR I=1 TO 10
30    READ A,B,C$
40    IF C$="-" THEN K=A-B:GOTO 80
50    IF C$="*" THEN K=A*B:GOTO 80
60    IF C$="/" THEN K=A/B:GOTO 80
70    IF C$="+" THEN K=A+B
80    PRINT A;C$;B"="
90    INPUT C
100   IF K=C THEN PRINT "セイカイ デ゛ス":S=S+1 ELSE
      PRINT "マチガ゛イ デ゛ス コタエハ";K:BEEP
110 NEXT I
120 PRINT "セイカイリツ ハ ";S/10*100;"% デ゛ス"
130 END
140 DATA 10,10,+
150 DATA 20,30,*
160 DATA 10,2,/
170 DATA 10,3,-
180 DATA 25,36,+
190 DATA 50,32,-
200 DATA 18,3,/
210 DATA 76,2,*
220 DATA 64,44,+
230 DATA 54,2,-
```

LESSON⑤

これでパソコンは暗算練習機

変数で飛びこし先を指定できる

次に、DATA（データ）文に書かれた問題を指定できるようにする形を考えてみましょう。それにはどうしたらよいでしょうか。DATA（データ）文に最初から順番に1、2、3、4……という番号がついているように想定して、数字を入力したらその番号の問題が表示されるようにすればよいのです。当然、くり返し命令は不要になります。

問題を指定するために加わる手順は、

①数字を入力して、問題を指定する。

②指定された問題を読みこむ。

の2つです。このうち、①についてはとくに問題はないでしょう。基本的には入力命令INPUT（インプット）を使えばよいのです。②は、①で入力された数値によって指定されたDATA（データ）文へ飛びこすようにします。これまで解説したなかでは判断命令IF～THEN（イフ・ゼン）を使えばよいのです。

Sが問題を指定する変数です。行200～220にDATA（データ）文（問題）があるとした場合を考えてみたものです。この場合、判断命令文が3行にもなります。この方式だと、DATA（データ）文の数だけ判断命令文が必要です。これを1行ですますことができる命令があります。それが、ゆく先指定命令ON～GOTO（オン・ゴーツー）です。

```
IF  S=1  THEN  200
IF  S=2  THEN  210
IF  S=3  THEN  220
```

ON（オン）のあとは「式」となっていますが、多くの場合、数値変数が入ります。また行番号は、コンマ（，）で区切ります。この命令を使うと、前述の3行が、1行ですみます。Sの値を1、2、3と変えていけば、飛びこし先の行番号を左からの順で指定することになります。

基本形

ON（オン）␣式␣GOTO（ゴーツー）␣行番号，行番号……

式や変数の値により、どこに飛びこすかを指定するゆく先指定命令

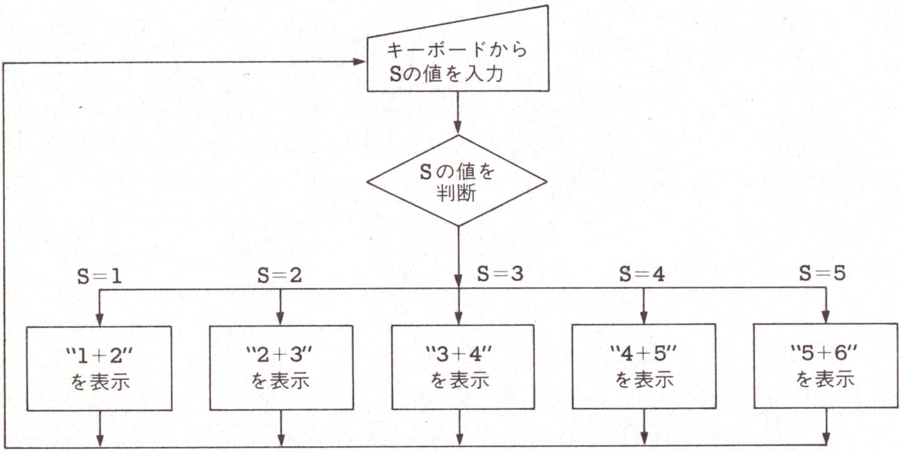
```
ON  S  GOTO  200,210,220
```

ゆく先指定命令をマスターしよう

簡単な例で、ゆく先指定命令 **ON～GOTO** を練習します。キーボードから入力するSの値を、いろいろ変えてみたのが実行結果です。Sの値が変化するにつれて、**GOTO** のあとの行番号が、左からの順番どおりに実行されていることがわかります（たとえば1を入力すれば行30）。

6を入力した場合のように、入力したSの値が、**GOTO** のあとの数より大きいときは、**GOTO** の次の行（行30）が実行されます。7でも8でも、結果は同じです。

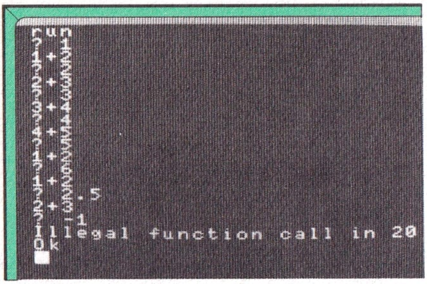
整数ではない数を入力すると、小数点以下が切りすてられて、その順番の行が実行されます（2.5なら2とみなされて行40）。また負の数を入力した場合は、「Illegal function call」（命令文の機能の呼び方が違う）というエラーメッセージが出ます。



●プログラム

```
10 INPUT S
20 ON S GOTO 30,40,50,
   60,70
30 PRINT "1+2":GOTO 10
40 PRINT "2+3":GOTO 10
50 PRINT "3+4":GOTO 10
60 PRINT "4+5":GOTO 10
70 PRINT "5+6":GOTO 10
```

実行



読みこむ DATA 文を指定する RESTORE

次に、前掲のプログラムに、読みこみ命令 **READ～DATA** を加えることを考えてみましょう。これまでの説明では、Sの値によって、問題（行番号）が指定できるはずです。ところが、実行すればわかるように、入力した数値に関係なく、行60～100の問題が、その順に表示されてしまいます。

●プログラム

```
10 INPUT S
20 ON S GOTO 60,70,80,90,100
30 READ A$
40 PRINT A$
50 GOTO 10
60 DATA "1+2":GOTO 30
70 DATA "2+3":GOTO 30
80 DATA "3+4":GOTO 30
90 DATA "4+5":GOTO 30
100 DATA "5+6":GOTO 30
110 END
```

DATA 文は、**ON～GOTO** では読みこまれず、「GOTO　30」で行30にいき、**READ** 文によってはじめて読みこまれますが、どの **DATA** 文を読みこむかは、**READ** 文で指定できないからです。それには、新しい命令が必要です。

基本形

RESTORE␣行番号

READ 文で読みこむ **DATA** 文を指定するデータ指定命令

RESTORE 文で行番号を指定すると、**READ** 文は指定された行番号の **DATA** 文から読みこみます。行番号を省略すれば、最初の **DATA** 文から読みこみを始めます。また、この命令を使用すると、同じデータを、何度でも使うことができます。**RESTORE** を使って、前出のプログラムを修正してみましょう。

●プログラム：RESTORE で修正したもの

```
10 INPUT S
20 ON S GOTO 60,70,80,90,100
30 READ A$
40 PRINT A$
50 GOTO 10
60 RESTORE 60:DATA "1+2":GOTO 30
70 RESTORE 70:DATA "2+3":GOTO 30
80 RESTORE 80:DATA "3+4":GOTO 30
90 RESTORE 90:DATA "4+5":GOTO 30
100 RESTORE 100:DATA "5+6":GOTO 30
110 END
```

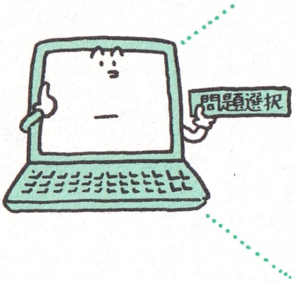
RESTORE 文は **DATA** 文の前において、マルチステートメントにもできます。

問題が選べる計算練習プログラム

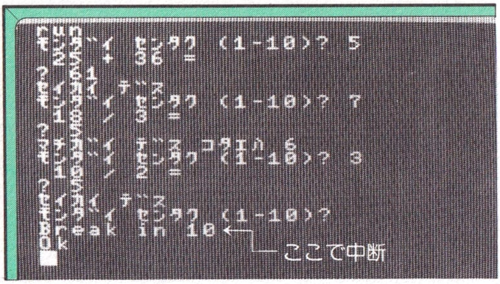
ゆく先指定命令 **ON～GOTO**（オン・ゴーツー）とデータ指定命令 **RESTORE**（リストア）をマスターしたところで、問題選択のできる計算練習プログラムを作成します。

●プログラム

```
10 INPUT "ﾓﾝﾀﾞｲ ｾﾝﾀｸ (1-10)";S
20 ON S GOTO 140,150,160,170,180,190,200,210,
   220,230
30 READ A,B,C$
40 IF C$="-" THEN K=A-B:GOTO 80
50 IF C$="*" THEN K=A*B:GOTO 80
60 IF C$="/" THEN K=A/B:GOTO 80
70 IF C$="+" THEN K=A+B
80 PRINT A;C$;B"="
90 INPUT C
100 IF K=C THEN PRINT "ｾｲｶｲ ﾃﾞｽ" ELSE PRINT "ﾏﾁ
    ｶﾞｲ ﾃﾞｽ ｺﾀｴﾊ";K:BEEP
110 GOTO 10
120 END
140 RESTORE 140:DATA 10,10,+:GOTO 30
150 RESTORE 150:DATA 20,30,*:GOTO 30
160 RESTORE 160:DATA 10,2,/:GOTO 30
170 RESTORE 170:DATA 10,3,-:GOTO 30
180 RESTORE 180:DATA 25,36,+:GOTO 30
190 RESTORE 190:DATA 50,32,-:GOTO 30
200 RESTORE 200:DATA 18,3,/:GOTO 30
210 RESTORE 210:DATA 76,2,*:GOTO 30
220 RESTORE 220:DATA 64,44,+:GOTO 30
230 RESTORE 230:DATA 54,2,-:GOTO 30
```



実行



乱数関数RND(ランダム)を使って次々に出題

最後に、パソコンが自分で問題をつくるプログラムを作成します。2ケタの加算で、正解率も表示して、出題を終えるかどうかを、「はい＝y」と「いいえ＝n」で指定できるようにします。

この課題でのポイントは、問題作成です。これには乱数関数**RND**(ランダム)を使います。2ケタまでの数値を発生させる乱数関数は、次のようになります（整数だけを扱うようにするため、整数関数も用いています）。

```
INT(RND(1)*99+1)
```

これを2つの変数（A, B）に代入して加える（A＋B）と加算の問題になります。また、そこで出題を終わらせるかどうかの判定は、判断命令**IF〜THEN**(イフ・ゼン)を使います。

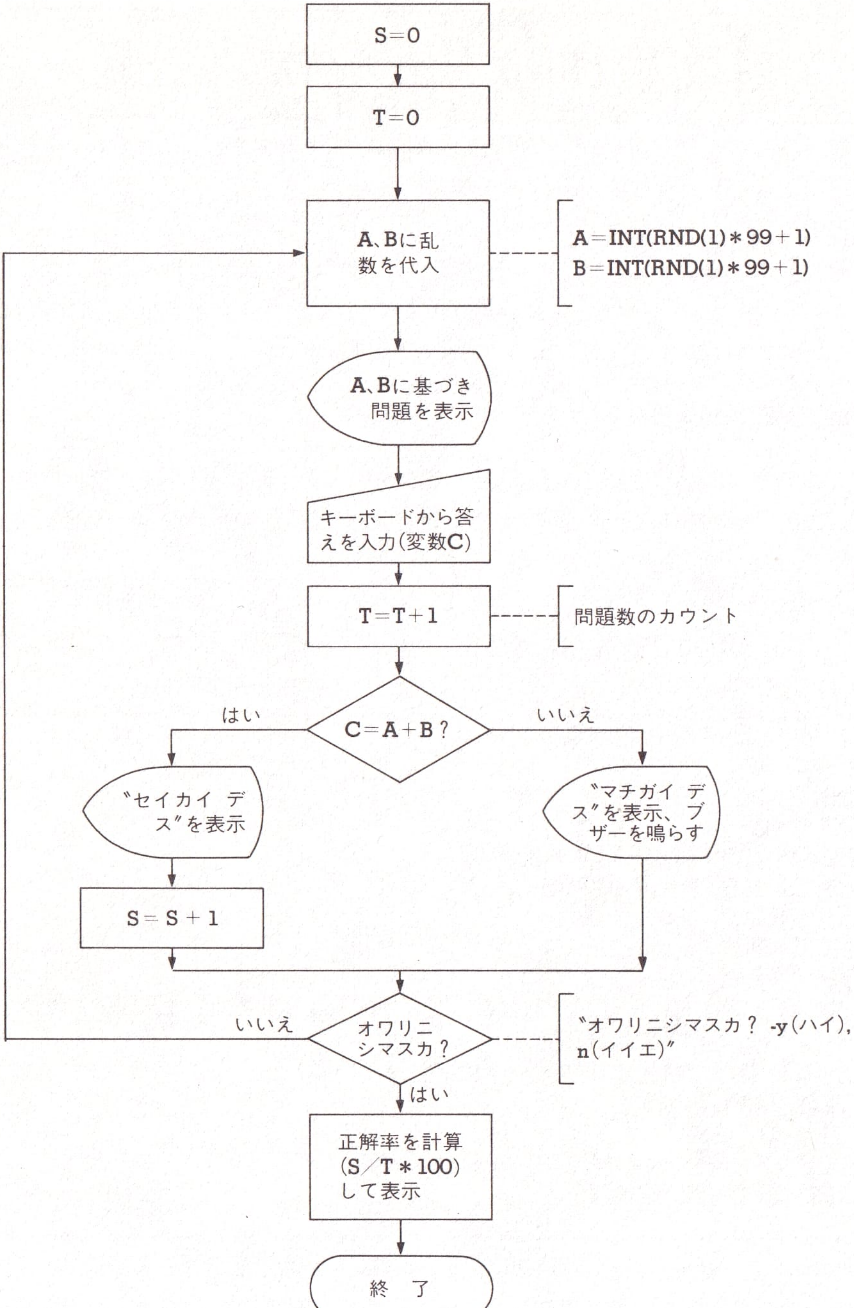
1問ごとに「オワリニシマスカ？」と表示されます。終わりにするときには[Y]、続けるときには[N]を押します。[Y]を押すと正解率が表示されて終わり、[N]を押すと、再び問題が表示されて続行されます。

●プログラム：パソコンが問題をつくる

```
10 S=0
20 T=0
30 A=INT(RND(1)*99+1)
40 B=INT(RND(1)*99+1)
50 PRINT A;"+";B
60 INPUT C
70 T=T+1
80 IF C=A+B THEN PRINT "ｾｲｶｲﾃﾞｽ":S=S+1 ELSE PRINT
   "ﾏﾁｶﾞｲﾃﾞｽ":BEEP
90 PRINT "ｵﾜﾘﾆｼﾏｽｶ?ｰy(ﾊｲ),n(ｲｲｴ)"
100 INPUT K$
110 IF K$="n" THEN 30
120 PRINT "ｾｲｶｲﾘﾂﾊ";(S/T*100)
130 END
```

練習問題8

上のプログラムを、加減乗除の別が選べるように修正してみよう。


S=0
T=0
A、Bに乱数を代入
A=INT(RND(1)*99+1)
B=INT(RND(1)*99+1)
A、Bに基づき問題を表示
キーボードから答えを入力(変数C)
T=T+1
問題数のカウント
C=A+B?
はい
いいえ
"セイカイ デス"を表示
S=S+1
"マチガイ デス"を表示、ブザーを鳴らす
オワリニシマスカ?
いいえ
はい
"オワリニシマスカ? -y(ハイ), n(イイエ)"
正解率を計算(S/T*100)して表示
終 了

ゲームで一服

スカッシュ・ゲーム

スカッシュというスポーツを知っているかな。壁に囲まれたコートでするテニスのようなものだ。

実行すると、黄色い壁ができて、右側には、赤いラケットが出てくる。このラケットをカーソル・キーで動かして、白いボールを打ち返そう。打ち返した回数が点数（SCORE）になる。

ただし、ボールは全部で5個しかない。1回打ちそこなうと1個減って、ボールがなくなれば、ゲーム終了だ。

別表のように、高い点になるほど、むずかしくなるからがんばってほしい。

このプログラムでジョイスティックを使えるようにするには、行470の「STICK」に続くカッコ内の値を1にすればいい。

6〜10点	中央にフィルターがかかり、その部分を通過するボールが見えなくなり、画面の上方まんなかに「ON」という表示が出る。フィルターは、2点得点するごとに広がる
11〜15点	コートの左のほうに障害物があらわれる
16〜20点	障害物が動きだす
21点以上	障害物の姿が見えなくなる

●プログラム

```
10 COLOR 15,4:SCREEN 2,2:WIDTH 29:KEY OFF
20 DEFINT A-Z:R=RND(-TIME)
30 DEF FNR(X)=RND(1)*X-1
40 OPEN"GRP:" FOR OUTPUT AS #1
50 '        ｽﾌﾟﾗｲﾄ ﾊﾟﾀｰﾝ ｾｯﾄ
60 FOR I=0 TO 2
70   S$=""
80   FOR J=1 TO 32
90     READ A$:S$=S$+CHR$(VAL("&h"+A$))
100    NEXT J
110    SPRITE$(I)=S$
120 NEXT I
130 '        ｼｮｷ ｼｮﾘ
140 PL=5:HT=0:BL=15:ZC=5
150 XB=96:YB=32:XR=230:YR=96:DX=8:DY=6:YA=50:YZ=96:DZ=6
160 CLS:PLAY "o4l64cdef"
170 '        ｶﾍﾞ ﾋｮｳｼﾞ
180 LINE(10,15)-(240,20),11,BF
190 LINE(10,175)-(240,180),11,BF
200 LINE(10,15)-(15,175),11,BF
210 LINE(16,21)-(240,174),5,BF
220 '        ｽﾌﾟﾗｲﾄ ﾜﾘｺﾐｻｷ
230 ON SPRITE GOSUB 530:SPRITE ON
240 '        ｼｮｷ ｽｺｱ ﾎﾞｰﾙ ﾋｮｳｼﾞ
250 PSET(30,5),4:PRINT #1,USING"Score:###";HT
260 PSET(150,5),4:PRINT #1,USING"Boll:##";PL
```

```
270 '         スプライト  ヒョウジ
280 PUT SPRITE 0,(XR ,YR),9,0
290 PUT SPRITE 1,(XB,YB),BL,1
300 IF TM<3 THEN TM=TM+1 ELSE SPRITE ON
310 IF HT>10 THEN PUT SPRITE 2,(YA,YZ),ZC,2
320 '         ボール ノ  セイギョ
330 XB=XB+DX:YB=YB+DY
340 IF XB<16 THEN PLAY "c":DX=-DX:XB=XB+DX
350 '         ボール  ヨケス
360 IF HT> 5  THEN PSET(120,5),4:PRINT #1,"on" ELSE
    GOTO 410
370 IF XB>50  THEN K=100-HT/2 ELSE BL=15:GOTO 410
380 IF XB =< K   THEN BL=15
390   IF XB > K   THEN BL=5
400     IF XB > 150 THEN BL=15
410 IF XB>250 THEN PLAY "bagfedcb":PL=PL-1:FOR W=0 TO
    3000:NEXT:IF PL=0 THEN GOTO 590 ELSE GOTO 150
420 IF YB<20 OR YB>165 THEN PLAY "G":DY=-DY:YB=YB+DY
430 '        ショウガイ  ブツ  セイギョ
440 IF HT=>0  AND HT<6  THEN GOTO 470 ELSE IF HT=>21
    THEN ZC=5 ELSE ZC=15
450 IF HT >15 THEN YZ=YZ+DZ: IF YZ<30 OR YZ>150 THEN
    PLAY "ACEG":DZ=-DZ:YZ=YZ+DZ
460 '        ラケット ノ  セイギョ
470 S=STICK(0)
480 Y=YR
490 IF S=1 THEN YR=YR-10
500 IF S=5 THEN YR=YR+10
510 IF(YR>5 )AND(YR<180) THEN GOTO 280 ELSE YR=Y:GOTO
    280
520 '         スプライト ワリコミ ショリ
530 SPRITE OFF:IF NOT(XB>30 AND XB<70) THEN PLAY "BE":
    HT=HT+1:GOSUB 560
540 DX=-DX:DY=(DY+YB-YR-10)/2+FNR(3):TM=0:RETURN 330
550 '         スコア ヒョウジ
560 LINE(80,5)-(105,13),4 ,BF
570 PSET(30,5),4:PRINT #1,USING"Score:###";HT:RETURN
580 '         シュウリョウ  ショリ
590 SCREEN 2,2:PSET (70,80),4:PRINT#1,"Game Over !!":
    PSET(100,100),4:PRINT#1,USING"Score:###";HT:FOR
    G=0 TO 15000:NEXT:CLOSE #1:END
600 '         キャラクタ パターン データ
610 DATA 1f,26,66,99,99,e6,e6,99,99,66,26,1f,00,00,00,
    00,80,40,60,90,90,70,70,90,90,70,58,f4,2a,15,0a,04
620 DATA 00,3c,7e,7a,7a,66,3c,00,00,00,00,00,00,00,00,
    00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
630 DATA 00,00,00,00,ff,00,ff,00,ff,00,ff,00,00,00,00,
    00,ff,00,ff,00,ff,00,00,00,00,00,ff,00,ff,00,ff,00
```

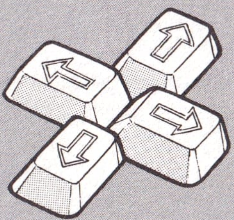
·PART·

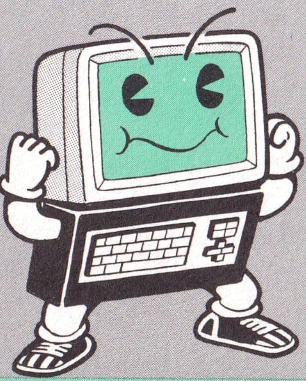
7

パソコンの"得意技"

データの並べかえ

FOR
〜
NEXT





住所録や名簿をつくるときには、あとで探しやすいようにアイウエオ順に並べるのがふつうです。しかし、一度にたくさんの名前を手ぎわよく並べかえるのは手間がかかります。これをパソコンにやらせてしまおう、というのが本章の課題です。

この章で学ぶ命令は、DIM(ディメンション)、SPC(スペース) MID$(ミドル・ダラー)、ASC(アスキー)、CLS(クリアスクリーン)、LEN(レングス)、CRH$(キャヤクタ・ダラー)、VAL(バリュー) です。

LESSON ①

たくさんのデータ処理は配列で

配列とはどういうことか

並べかえのプログラムにとりかかる前に、「配列」ということについて理解しておいていただきたいと思います。配列の考え方は、同じような性質や大きさを持つデータを、数多く処理するときに便利です。

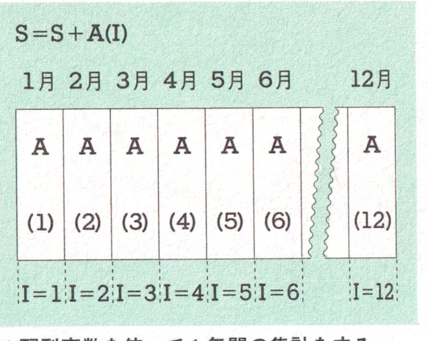
配列を構成する変数を配列変数といい、A(5)、AB(10) などのように、変数名にカッコをつけて書きます。カッコ内の数字を添え字といいます。

配列変数は、たとえばAという1つの変数名で、データの収納場所(記憶場所)をいくつも用意することができる、という利点があります。たとえば、毎月の売りあげ合計を1年ぶん集計する場合を考えてみましょう。

各月の売りあげを入力して集計するとなれば、変数は月ごとに1つ必要ですから、たとえば1月ぶんはA、2月ぶんはBといったぐあいに、A～Lまで12個の変数を用意しなくてはなりません。ところが、配列変数を使えば、Aという変数だけですみます。Aのあとに添え字をつけて、その値を1～12まで変化させれば、1年ぶんの売りあげをすべておさめることができるのです。

この場合、添え字のほうも変数にして、A(I) という形にします。それで、Iの値を1～12まで変化させればよいのです。1年間の集計をSとすれば、下の図式のようなしくみになります。

Iを1～12まで連続的に変化させていくには、くり返し命令FOR～NEXT(フォー・ネクスト)を使うことが予想できると思います。配列変数の便利さのひとつは、このくり返し命令で処理できるという点にあるのです。しかも、添え字の値はひとつの変数名でRAM(ラム)の容量いっぱいまで可能ですから、似たような一連のデータを多量に扱うときにも便利です。



▲配列変数を使って1年間の集計をする

配列宣言でデータの収納場所を確保

プログラムの中で配列変数を使うためには、どんな配列変数名にするか、同じ変数名の収納場所（配列要素）をいくつ用意するかなどについて、パソコンに予約しておく必要があります。予約には配列宣言命令 **DIM**（ディメンション）を使います。

たとえば「DIM　A(10)」とすると、A の配列変数が A(0)～A(10) まで、11 個用意されることになります。個々の配列要素はいわば記憶場所で、配列宣言命令が実行されるとその場所が確保され、入力されたデータが収納されます。

ひとつの **DIM**（ディメンション）文で、複数の配列変数を宣言することもできます。その場合には、

```
DIM  A(10), B(5), C$(50), D$(25)
```

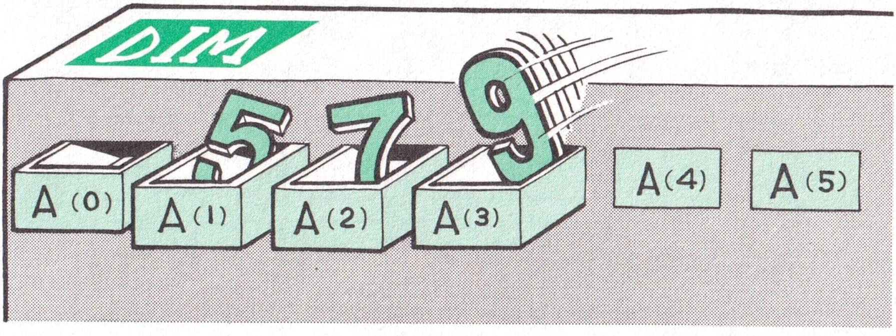
のようにコンマ（，）で区切ります。

プログラム中で、宣言した添え字の最大値よりも大きい値の添え字が使われると、「Subscript out of range」（サブスクリプト アウト オブ レインジ）（配列変数の添え字の値が規定範囲外）というエラーメッセージが出ます。また、配列変数を配列宣言命令なしで使うと、自動的にその変数の添え字の最大値は 10 となり、11 個の変数が用意されます。

基本形

DIM（ディメンション）**␣変数名（添え字の最大値），……**

配列変数名と配列要素の数を宣言し、パソコン内にその配列の領域を確保する配列宣言命令



簡単な例で配列を使ってみよう

では、配列がどんなものか、実際に使ってみることにしましょう。次のプログラムを入力し、実行してみてください。

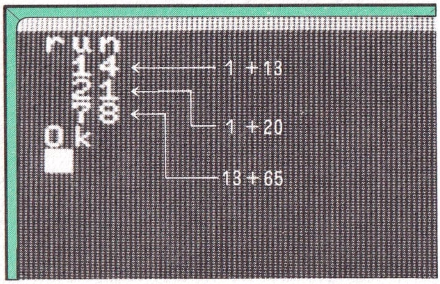
行 10 で A(10) と宣言していますから、配列変数は A(0) から A(10) までの 11 個用意されています。そして、行 20 によって、A(0) には 1、A(5)には 13 がしまいこまれました。ほかの配列変数は中身がありません。

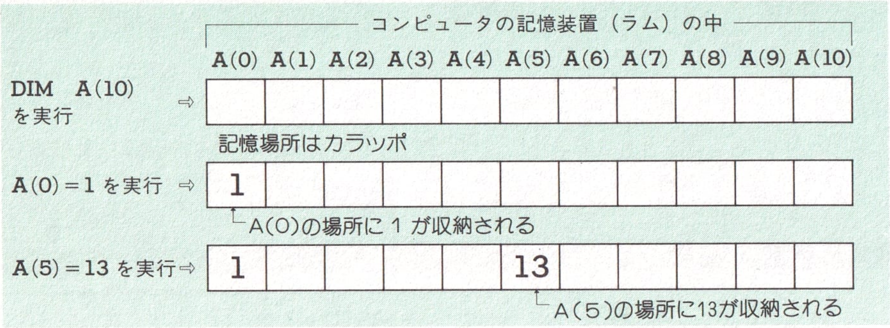
行 30 から行 50 では、それぞれの配列変数を使って計算をやらせています。なぜ計算の結果がこうなったか考えてください。

●プログラム

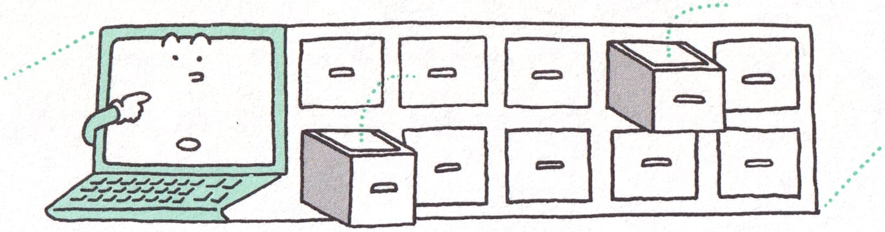
```
10 DIM A(10)
20 A(0)=1:A(5)=13
30 PRINT A(0)+A(5)
40 PRINT A(0)+20
50 PRINT A(5)+65
60 END
```

実行





▲「DIM A(10)」を実行



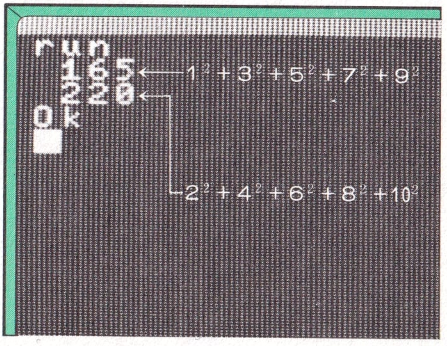
データの収納はくり返し命令で

配列宣言命令 **DIM** 文は、くり返し命令 **FOR～NEXT** とともに使うと便利です。プログラムの行 10～40 により、I の値が 1、2、3、4……と変わっていくにつれて、配列変数 A(1)～A(10) に 1(1∧2)、4(2∧2)、9(3∧2)、16(4∧2) ……が代入されていきます。

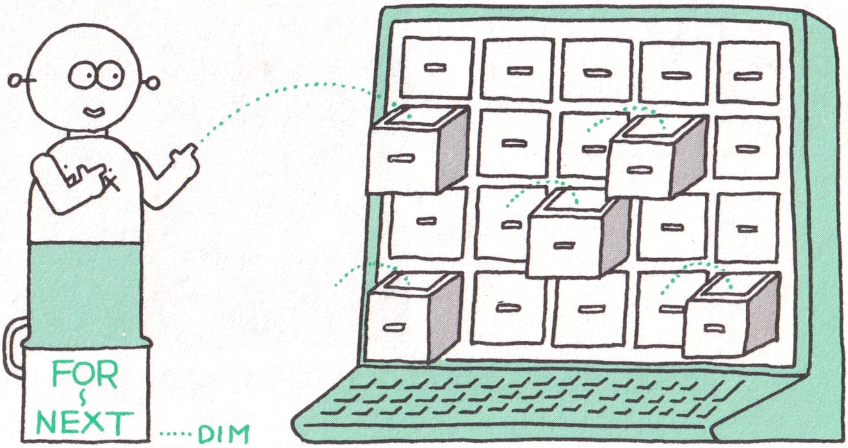
行 50～80 では、指定した配列変数の加算を行なっています。A の値には奇数の 2 乗、B の値には偶数の 2 乗が入っていて、それぞれを加算していくことになります。

●プログラム

```
10 DIM A(10)
20 FOR I=1 TO 10
30    A(I)=I^2
40 NEXT I
50 FOR I=1 TO 9 STEP 2
60    A=A+A(I)
70    B=B+A(I+1)
80 NEXT I
90 PRINT A
100 PRINT B
110 END
```



実行



入力命令INPUTでさらに多くのデータを

ややこしい累乗計算が、いとも簡単にできました。このような芸当は、配列変数ならではのことです。配列変数を使わないで同じことをしようとすると、もっと複雑なプログラムになります。

入力命令 **INPUT** を使ってたくさんのデータを順番に入力する場合にも、配列変数が威力を発揮します。例にあげたのは、10個の数を入力して、入力が終わると合計を表示するプログラムです。くり返し命令によって、配列変数A(I)の添え字Iの値を1～10に変化させることで、入力したデータを順番に収納しています。

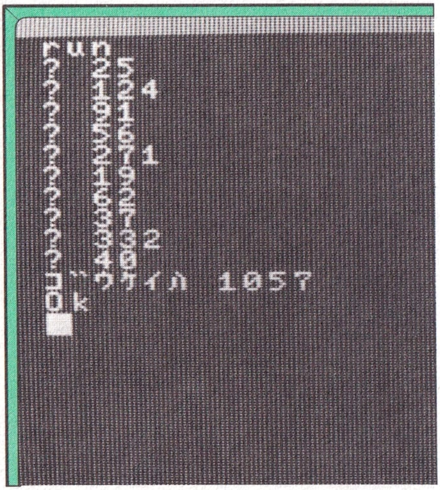
行10ではA(0)～A(10)の11個の配列要素を用意したのに、実際にはA(0)は使われていません。くり返し命令の初期値が1から始まっているからです。このような使い方もあります。

このようなプログラムをつくるときには、配列宣言命令の添え字の値は多めに用意しておくほうが、不都合が起こりません。前述したように、くり返し命令の終値が添え字の最大値を超えると、エラーメッセージが出て、実行がストップしてしまうからです。

●プログラム：入力した10個の数を合計

```
10 DIM A(10)
20 FOR I=1 TO 10
30    INPUT A(I)
40    S=S+A(I)
50 NEXT I
60 PRINT "ｺﾞｳｹｲﾊ";S
70 END
```

➔ 実行



練習問題9

本項のプログラムを修正して、入力したデータがすべて式として表示されるようにしよう。

LESSON ②

こうすればデータ処理はもっと便利

配列の大きさを自由にできると応用がきく

実際になんらかの仕事にパソコンを使う場合には、入力するデータの数は一定していないのがふつうです。配列要素の数は毎回違うのですから、最初にその数を宣言してしまうと、ぐあいの悪いことが出てきます。

そこで、データの数によって配列要素の数（添え字の最大値）を決められるプログラムが考えられます。それには、最初にデータの数を入力できるようにし、その入力した値を配列宣言命令 **DIM**（ディメンション）の添え字の最大値にします。また、くり返しの回数も同じにします。つまり、この3つの変数を同じにすればよいのです。

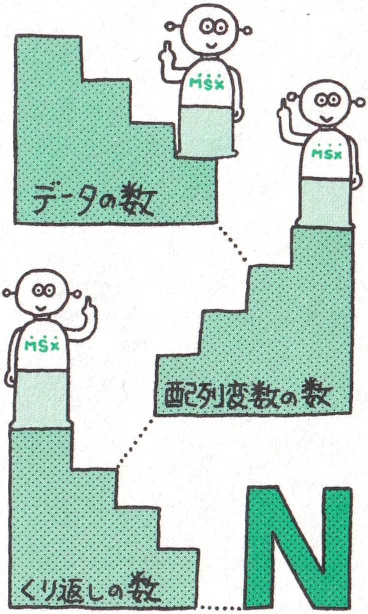
下のプログラムでは、データ数の入力命令の変数、配列宣言命令の添え字、くり返し命令の終値にすべて「N」という変数を使っています。

●プログラム：入力した数をすべて合計

```
10 INPUT "データノカズハ";N
20 DIM A(N)
30 FOR I=1 TO N
40     INPUT A(I)
50     S=S+A(I)
60 NEXT I
70 PRINT "ゴウケイハ";S
80 END
```

実行

```
run
データノカズハ? 5
? 90
? 2050
? 220
? 715
? 182
ゴウケイハ 3257
Ok
```



データの内容が書けるとこづかい帳に

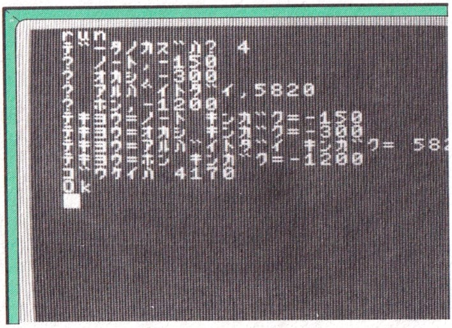
いくつものデータが簡単に処理できるようになると、テストの結果を集計したり、こづかい帳をつけたりすることができるようになります。たとえば、こづかい帳をつくる場合を考えてみましょう。記入するのは、大まかにいって収入と支出です。ふつうは収入と支出を別々にしないで、使ったりもらったりした順番に書いていきます。そして、残金（残ったお金）を計算するとき、収入は足し算をし、支出は引き算をすることになります。

そこで、支出には金額の前にマイナス記号をつけます。こうすれば、収入と支出を一度に計算（加算）できます。また、支出や収入がどんなものなのかを記入できるようにすると便利です。

前出のプログラムを修正、追加して下のようなプログラムをつくってください。

行20と行40にある「T$(　)」は、データの内容を代入するための変数です。データの内容は文字ですから、ダラー記号$をつけて文字変数にしてあります。「$」を忘れて文字を代入しようとすると、エラーメッセージが出ます。

入力する際には、摘要（内容）と金額の順に、収入は金額をそのまま、支出はマイナス記号(－)をつけます。



実行

●プログラム：こづかい帳

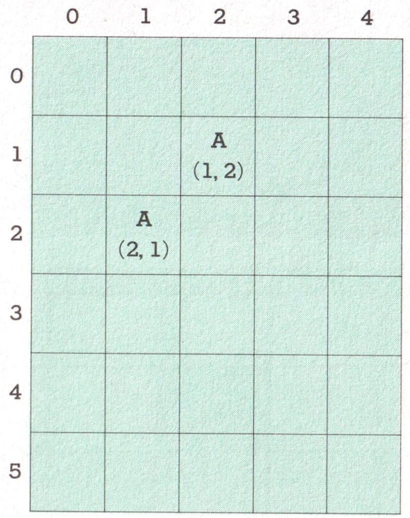
```
10 INPUT "ﾃﾞｰﾀﾉｶｽﾞﾊ";N
20 DIM T$(N),A(N)
30 FOR I=1 TO N
40    INPUT T$(I),A(I)
50    S=S+A(I)
60 NEXT I
70 FOR I=1 TO N
80    PRINT "ﾃｷﾖｳ=";T$(I);" ｷﾝｶﾞｸ=";A(I)
90 NEXT I
100 PRINT "ｺﾞｳｹｲﾊ";S
110 END
```

添え字を増やせば配列はさらに便利に

ここまで説明してきた配列変数は、すべて添え字がひとつでした。このような配列を1次元配列といいます。配列にはこのほか、複数の添え字を持つ多次元配列もあります。多次元配列のうち、2次元配列は、「DIM　A(5, 4)」のように宣言します。配列変数の添え字はコンマ（ , ）で区切ります。

この **DIM**（ディメンション）文によってできる記憶の箱は、縦、横に並んでいると考えることができます。縦は0～5の6つ、横は0～4の5つの箱で、合計30の箱がアパートのように積み重なっているわけです。それぞれの箱はA(2, 1)、A(1, 2)という形で区別されています。さて、このように並んだ箱の中に、下図のようにカタカナを収納し、指定した箱から呼びだすプログラムをつくります。箱の番号は、それぞれ1～6、1～5と変えます。

カタカナを収納するには、読みこみ命令 **READ～DATA**（リード・データ）を使うことにします。プログラムを見ると、FOR-NEXT（フォー・ネクスト）ループが二重に使われています。内側のループから処理されますから、最初はア～オと横の列が読みこまれ、それが終わると1段さがり（Iが1増えて）、また横に読みこまれます。データの読みこみは行60までで終わり、行70～90で指定した場所のカタカナを表示します。



▲DIM　A(5, 4)のときの2次元配列

J →, I ↓

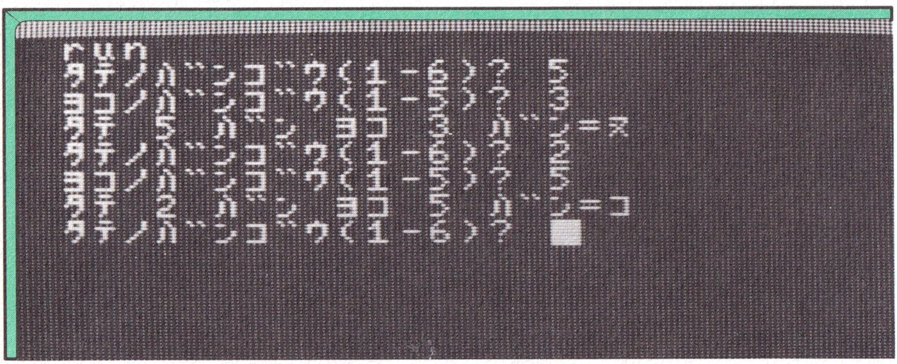
I＼J	1	2	3	4	5
1	ア	イ	ウ	エ	オ
2	カ	キ	ク	ケ	コ
3	サ	シ	ス	セ	ソ
4	タ	チ	ツ	テ	ト
5	ナ	ニ	ヌ	ネ	ノ
6	ハ	ヒ	フ	ヘ	ホ

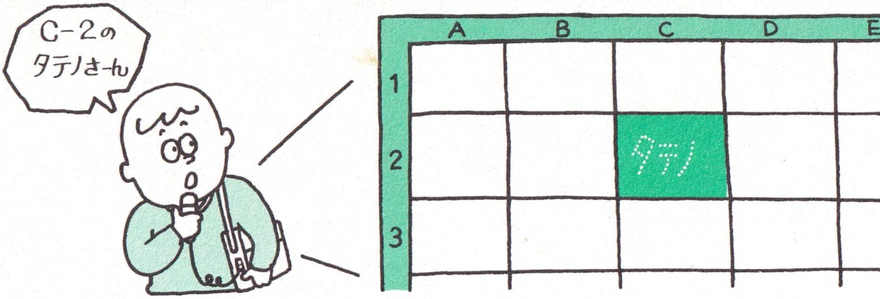
▲カタカナを収納する

●プログラム

```
10 DIM A$(6,5)
20 FOR I=1 TO 6
30    FOR J=1 TO 5
40       READ A$(I,J)
50    NEXT J
60 NEXT I
70 INPUT "ﾀﾃﾉﾊﾞﾝｺﾞｳ(1-6)";I
80 INPUT "ﾖｺﾉﾊﾞﾝｺﾞｳ(1-5)";J
90 PRINT "ﾀﾃ";I;"ﾊﾞﾝ ﾖｺ";J;"ﾊﾞﾝ=";A$(I,J)
100 GOTO 70
110 DATA ｱ,ｲ,ｳ,ｴ,ｵ
120 DATA ｶ,ｷ,ｸ,ｹ,ｺ
130 DATA ｻ,ｼ,ｽ,ｾ,ｿ
140 DATA ﾀ,ﾁ,ﾂ,ﾃ,ﾄ
150 DATA ﾅ,ﾆ,ﾇ,ﾈ,ﾉ
160 DATA ﾊ,ﾋ,ﾌ,ﾍ,ﾎ
170 END
```

実行





入力したデータを消すには空白関数

配列を使って、人名を入力して記憶するプログラムをつくりましょう。手順はフローチャートのようになります。このうち、③の処理を行なっているのは、画面を見やすくするためです。この処理を行なうものとして話を進めます。

行60に新しい命令**SPC**（スペース）が使われています。基本形のカッコ内には、数値のほか、変数や数式を入れることもできます。ただし数値は、0～255の範囲でなければなりません。行60の「SPC(20)」は、20字ぶんの空白を表示しているわけです。その位置は、氏名の入力された位置と同じ「10, 1」ですから、この行によって、入力された氏名を消していることがわかります。

このプログラムでは、配列変数の配列要素を11個しか宣言していません（行10）が、入力する氏名の数を多くする場合には、カッコ内の数値を大きくします。

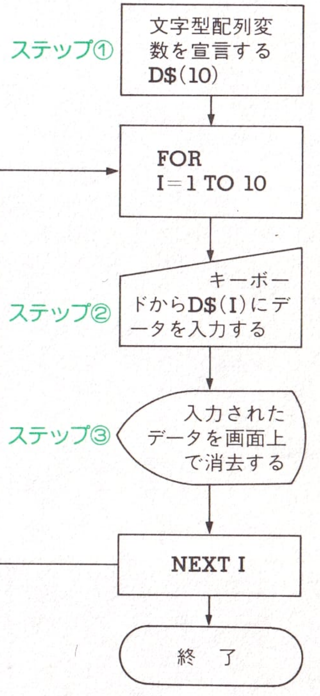
●プログラム

```
10 DIM D$(10)
20 FOR I=1 TO 10
30     LOCATE 10,1
40     INPUT "ｼﾒｲ";D$(I)
50     LOCATE 10,1
60     PRINT SPC(20)
70 NEXT I
80 END
```

基本形

SPC（スペース）（数値など）

指定した数だけ空白を表示する空白関数



（注） LOCATEによる表示位置指定は、図が複雑になるため省略（以下、同）

練習問題10

本項のプログラムを、最初にデータの数を入力するよう修正してみよう。

配列への収納、画面への表示は入力順

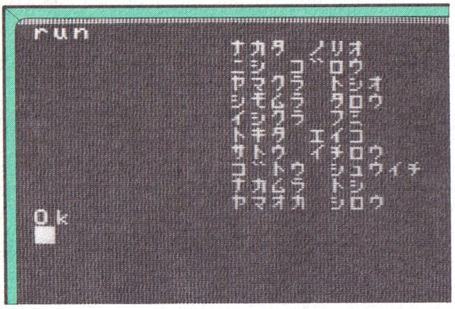
入力した氏名を一度、すべて画面に表示するプログラムにします。手順としては、「D$(I) の内容を画面に表示する」が加わるだけです（行 80～100)。このプログラムを実行し、たとえば実行画面のような氏名を入力すると、それらは上から順に D$(1)～D$(10) に収納されます。そして、この氏名リストを 50 音順に並べるのが、本章の主題です。

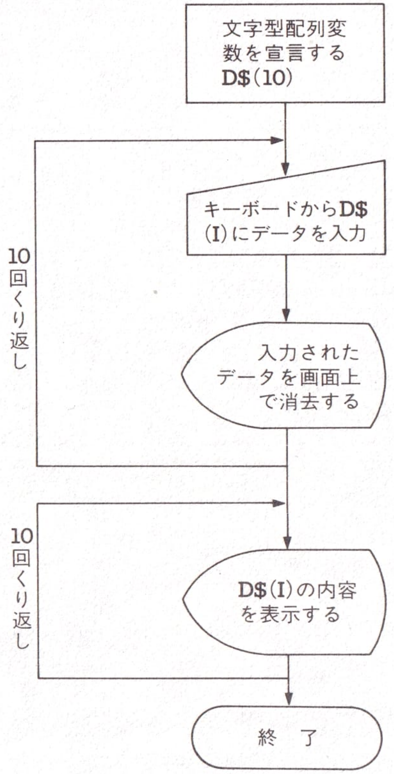
でたらめに並んでいるデータを、ある基準に従って一定の順序に並べかえることを、分類あるいはソートといいます（本書では、並べかえと呼んでいます)。並べかえは、クラス名簿、住所録など、実生活で数多く見られる作業です。このような仕事をさせるプログラムをつくれば、さらに、パソコンの活用範囲が広がります。

●プログラム

```
10 DIM D$(10)
20 FOR I=1 TO 10
30     LOCATE 10,1
40     INPUT "ｼﾒｲ";D$(I)
50     LOCATE 10,1
60     PRINT SPC(20)
70 NEXT I
80 FOR I=1 TO 10
90     LOCATE 10,I
100    PRINT D$(I)
110 NEXT I
120 END
```

実行





（注）「くり返し」としてある部分にはFOR～NEXTが使われている。

LESSON—③

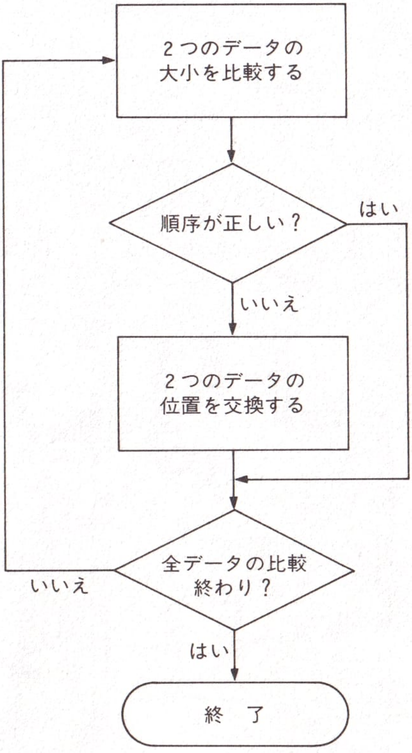
並べかえの基本をマスターしよう

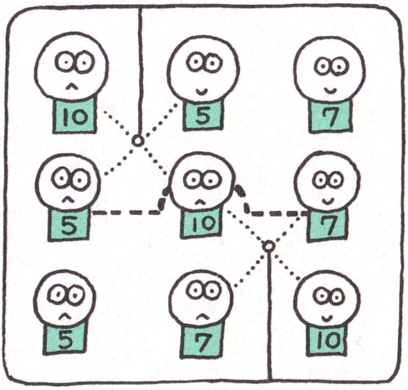
2つのデータ比較が並べかえの基本

並べかえの方法はいろいろありますが、もっとも一般的なのは、たくさんあるデータの中からとり出した2つのデータを比較し、その大小によって順番を入れかえる作業をくり返す方法（交換分類法）です。これにより、すべてのデータの比較と並べかえを行ないます。

たとえば、「10、5、7」というデータを小さい順に並べかえるとすると、まず10と5を比べ、5のほうが小さいので「5→10」と位置を入れかえます。次に、5と7を比べますが、5のほうが小さいので位置はそのまま。さらに、10と7を比べます。こんどは7のほうが小さいので「7→10」と並べかえます。これで「5→7→10」と、最初の目的どおり、数の小さい順に並びました。

この方法は単純でわかりやすいのですが、データが多いときには手間がかかるのが欠点です。





仮に入れておく変数が必要

プログラムをつくるうえでポイントになるのは、2つのデータの位置を交換する方法です。変数の値は移動すると古い値が消えてしまうので、図のような方法をとります。小さい順に並べる場合を例にします。

図で、変数Xは、交換する変数（ここではA）の値を一時保管しておく、待機用の変数です。①〜③を代入命令文で書くと、次のようになります。

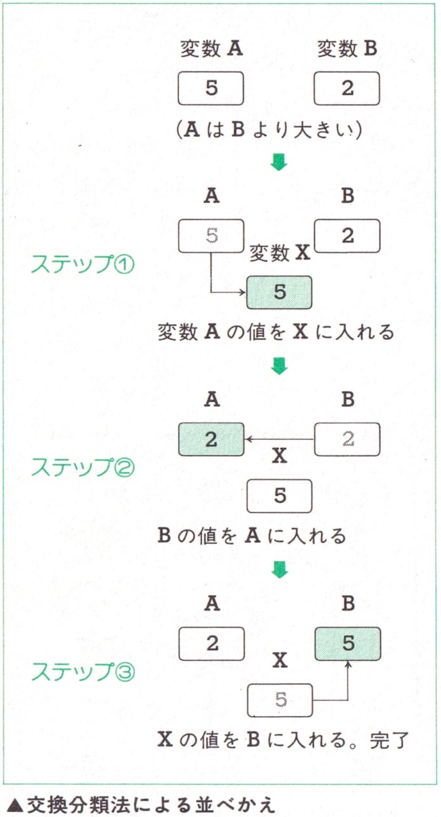
①X＝A

②A＝B

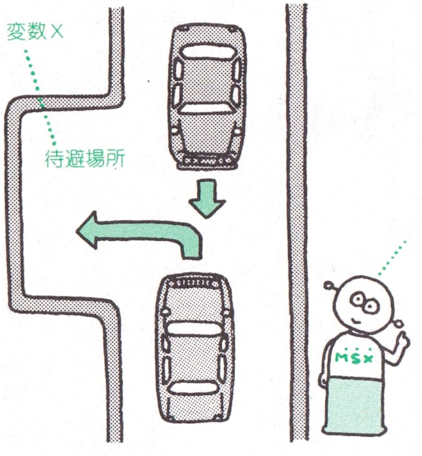
③B＝X

交換分類法で並べかえのプログラムを作成するときには、特別にこの待機用変数を用意する必要があります。

この方法は、車が1台ぶんしか通れないような細い道を、2台の自動車がすれ違うことによく似ています。そのままでは位置を入れかえることができないので、車がよけられるような場所をつくっておいて、どちらか一方の車がそこに入り、他方の車を先に通します。この待避場所が、変数Xと考えればよいでしょう。



▲交換分類法による並べかえ



数を一定のルールで並べかえよう

では、5個の数を入力し、それを小さい順に並べかえるプログラムをつくってみましょう。小さい順に並べることを「昇順」、大きい順に並べることを「降順」といいます。くわしい説明は次項にゆずるとして、まず下のプログラムを見てください。その構成は次のようになっています。

①データの入力――行10～40

②データの並べかえ――行60～110

③並べかえたあとのデータ表示――行120～140

並べかえ処理の中心は、行90です。最初の「IF　A(I)＞A(J)」は、「もしA(I)がA(J)より大きければ」という意味です。「＞」は、両側の2つのデータの関係をあらわす記号（関係演算子）です。ここで、A(I)とA(J)が比較され、大小関係が判断されます。

この行が実行されるとA(I)とA(J)の比較が行なわれて、もしA(I)のほうが大きければ、位置を交換することになります。それが、同じ行の後半にある「THEN W＝A(I)：A(I)＝A(J)：A(J)＝W」です。これは、前述した3つの手順をマルチステートメント（多重命令）にまとめたもので、Wが待機用変数です。

●プログラム：5つの数を小さい順に並べかえ

```
10 DIM A(5)
20 FOR I=1 TO 5
30     INPUT A(I)
40 NEXT I
50 CLS
60 FOR I=1 TO 4
70     JS=I+1
80     FOR J=JS TO 5
90         IF A(I)>A(J) THEN W=A(I):A(I)=A(J):
           A(J)=W
100    NEXT J
110 NEXT I
120 FOR I=1 TO 5
130     PRINT A(I)
140 NEXT I
150 END
```

くり返し命令によるFOR-NEXT(フォー・ネクスト)ループが、二重に使われています。外側のループで、その配列を順に指定していき、内側のループで、その配列変数とそれ以降の配列変数を比較・並べかえしています。つまり、まずA(1)を基点にして、その他のすべての変数を比較します。その過程で、昇順ではない組み合わせは並べかえが行なわれます。したがって、外側のループが最初に1回終わると、A(1)にはデータの最小値が入ります。

A(1)のデータはこれで決まりです。次にA(2)を基点に、それ以降の変数との比較・並べかえを行ないます。これにより、A(2)には最小から2番めの値が入ります。

こうした処理をくり返すと、A(4)が決まった時点でA(5)には自動的に最大値が入ります。このため、くり返しの回数は、「データ数マイナス1回」ですみます。行60のくり返し命令の終値が「4」になっているのは、このためです。

C/O/L/U/M/N

◉2進数は16進数にしてわかりやすく◉

前に、コンピュータの世界は白と黒(0と1)の2種類しかないことは説明した(60ページ)。つまり、2進数の世界である。けれども、0と1ばかり並んだ数字は、人間にはたいへん扱いにくい。書くのもめんどうだし、ひと目見てわかりにくい。機械にとって都合のよいことも、人間にとっては不都合というわけ。

そこで、実際にコンピュータに伝えるときには、2進数を16進数に変換してあらわすことになっている。

16進数とは、ことばのとおり16で1ケタくりあがる数のかぞえ方だ。10進数や2進数と対比すると、次のようになる。

これだと、2進数の4ケタぶんを1ケタであらわせるから、それだけ簡単になるわけだ。

10進数	2進数	16進数
0	00000000	0
1	00000001	1
2	00000010	2
3	00000011	3
4	00000100	4
5	00000101	5
6	00000110	6
7	00000111	7
8	00001000	8
9	00001001	9
10	00001010	A
11	00001011	B
12	00001100	C
13	00001101	D
14	00001110	E
15	00001111	F
16	00010000	10
17	00010001	11

並べかえは単純な作業のくり返し

前出のプログラムを実行し、7、5、10、1、3のデータをこの順番に入力したとします。並べかえは、下図のように行なわれます。図で、「⎵」は比較で、「⎵×」は並べかえです。外側のループがくり返されるほど、内側のくり返しの回数は少なくなります。データの数がどんなに多くても、原理は同じです。

ところで、2つのデータを比較して大小関係を判断するには、前述した関係演算子を使います。関係演算子の両側に比較するデータを書いたものが「関係式」です。比較するデータは、定数でも変数でも、あるいは数式でもよく、また、数値でも文字でも比較できます。しかし、数値と文字を比べることはできません。

関係式と関係演算子には、別表のようなものがあります。表中、「＜」や「＞」などが関係演算子で、その基本は「＜」「＞」「＝」の3種類です。このうち「＝」は代入文でも使いますが、意味は違うので注意します（117ページ参照）。



◉関係式と関係演算子

関係式の例	意　味
A＝B	AはBと等しい
A＜B	AはBより小さい（AはB未満）
A＞B	AはBより大きい
A＜＝B A＝＜B	AはBより小さいか等しい （AはB以下）
A＞＝B A＝＞B	AはBより大きいか等しい （AはB以上）
A＜＞B A＞＜B	AとBは等しくない

練習問題11

入力した数値15個を、降順に並べかえるプログラムをつくってみよう。

LESSON ④

文字データは数値に変換して並べかえ

入力した氏名も即座にアイウエオ順に

入力するデータが文字であっても、比較・並べかえは数値と同じように行なえます。「でたらめに名前を入力し（カタカナ）、アイウエオ順に並べかえて表示する」という課題を考えてみましょう。ここで注意したいのは、変数が文字型変数である、ということです。したがってダラー記号（$）のついた変数にしなければなりません。

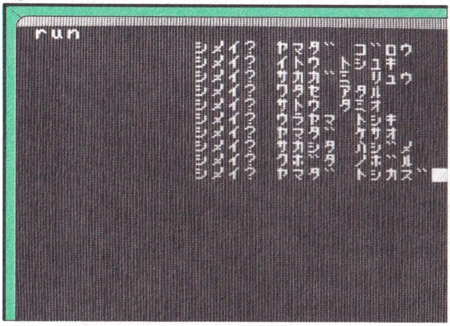
正しく入力できたら実行してみます。全データの入力が終わると、画面のデータは消え、並べかえたデータが表示されます。

フローチャートを見ると、さまざまな作業がどのように組み立てられているか、だいたいのところが把握できるはずです。

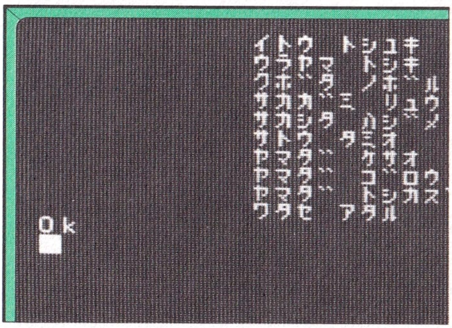
●プログラム：名前を50音順に並べかえ

```
10 DIM D$(10)
20 FOR I=1 TO 10
30    LOCATE 10,I
40    INPUT "ｼﾒｲ";D$(I)
50 NEXT I
60 CLS
70 FOR I=1 TO 9
80    JS=I+1
90    FOR J=JS TO 10
100      IF D$(I)>D$(J) THEN W$=D$(I):D$(I)=D$
         (J):D$(J)=W$
110   NEXT J
120 NEXT I
130 FOR I=1 TO 10
140    LOCATE 10,I
150    PRINT  D$(I)
160 NEXT I
170 END
```

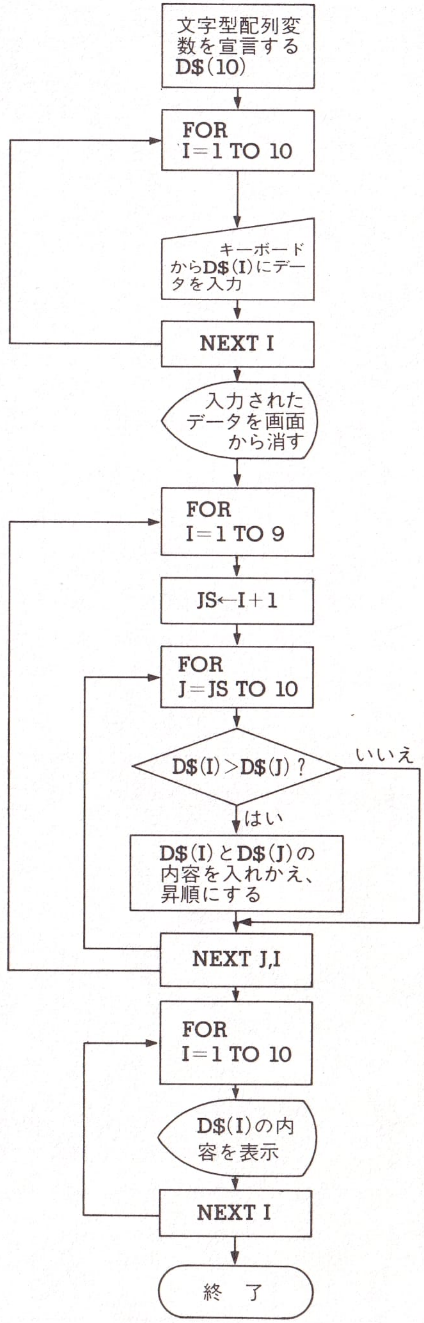
入力が終わったところ



50音順に表示







文字データはコードに変える

文字データの比較・並べかえができるのは、それぞれの文字に固有の番号がつけられていて、パソコンは内部でその番号によって文字を扱っているからです。入力した文字はすべて数字におきかえられて、記憶や処理が行なわれているのです。

この番号のことを「キャラクタコード」といいます。これを一覧表にしたのがキャラクタコード表です。これを見ると、最初が0、Aが65、aが97です。つまり、下にいくほど、また右へいくほど、数が大きくなります。

カタカナは、「ア」が177で、以下順に1ずつ増えていき、「ン」は221になります。コンピュータはこの数字によって判断するので、結局のところ、アイウエオ順の並べかえは、数値を昇順に並べかえるのと同じことです。

氏名などのような文字列でも、それぞれの文字をすべて数値におきかえて処理されます。文字列全体をひとつのまとまりとして数値にし、それを比較することによって、並べかえの必要があるかどうかを判断することになります。

キャラクタコード表を見れば、アルファベットもABC順に、ひらがなも50音順に並べかえられることが理解できるでしょう。

◉グラフィックキャラクタコード表

キャラクタコード表とグラフィックキャラクタコード表は、似ていますが、出力の方法などが違います。別々の2枚の表であると考えたほうがよいでしょう。2つの表にあるキャラクタは、すべてキーボードから打ちこむことができます。

コード(10進)	CHR	コード(10進)	CHR	コード(10進)	CHR	コード(10進)	CHR	コード(10進)	CHR	コード(10進)	CHR
0		6	土	12	秒	18	┬	24	┌	30	中
1	月	7	日	13	百	19	┤	25	┐	31	小
2	火	8	年	14	千	20	├	26	└		
3	水	9	円	15	万	21	┼	27	┘		
4	木	10	時	16	π	22	│	28	╳		
5	金	11	分	17	┴	23	─	29	大		

◉キャラクタコード表

数字はキャラクタコード、CHR 欄の文字や記号はキャラクタといいます。0～31 のコントロールキャラクタとは、カーソルや画面の操作を行なうキャラクタのことです。

コード (10進)	CHR	コード (10進)	CHR	コード (10進)	CHR	コード (10進)	CHR	コード (10進)	CHR	コード (10進)	CHR
0	↑	43	+	86	V	129	♥	172	ャ	215	ラ
1		44	,	87	W	130	♣	173	ュ	216	リ
2		45	−	88	X	131	♦	174	ョ	217	ル
3		46	.	89	Y	132	○	175	ッ	218	レ
4		47	/	90	Z	133	●	176	ー	219	ロ
5		48	0	91	[	134	を	177	ア	220	ワ
6		49	1	92	¥	135	ぁ	178	イ	221	ン
7		50	2	93	]	136	ぃ	179	ウ	222	゛
8		51	3	94	^	137	ぅ	180	エ	223	゜
9		52	4	95	_	138	ぇ	181	オ	224	た
10		53	5	96	‘	139	ぉ	182	カ	225	ち
11		54	6	97	a	140	ゃ	183	キ	226	つ
12		55	7	98	b	141	ゅ	184	ク	227	て
13		56	8	99	c	142	ょ	185	ケ	228	と
14		57	9	100	d	143	っ	186	コ	229	な
15	コントロールキャラクタ	58	:	101	e	144		187	サ	230	に
16		59	;	102	f	145	あ	188	シ	231	ぬ
17		60	<	103	g	146	い	189	ス	232	ね
18		61	=	104	h	147	う	190	セ	233	の
19		62	>	105	i	148	え	191	ソ	234	は
20		63	?	106	j	149	お	192	タ	235	ひ
21		64	@	107	k	150	か	193	チ	236	ふ
22		65	A	108	l	151	き	194	ツ	237	へ
23		66	B	109	m	152	く	195	テ	238	ほ
24		67	C	110	n	153	け	196	ト	239	ま
25		68	D	111	o	154	こ	197	ナ	240	み
26		69	E	112	p	155	さ	198	ニ	241	む
27		70	F	113	q	156	し	199	ヌ	242	め
28		71	G	114	r	157	す	200	ネ	243	も
29		72	H	115	s	158	せ	201	ノ	244	や
30		73	I	116	t	159	そ	202	ハ	245	ゆ
31	↓	74	J	117	u	160		203	ヒ	246	よ
32		75	K	118	v	161	。	204	フ	247	ら
33	!	76	L	119	w	162	「	205	ヘ	248	り
34	"	77	M	120	x	163	」	206	ホ	249	る
35	#	78	N	121	y	164	、	207	マ	250	れ
36	$	79	O	122	z	165	・	208	ミ	251	ろ
37	%	80	P	123	{	166	ヲ	209	ム	252	わ
38	&	81	Q	124	¦	167	ァ	210	メ	253	ん
39	'	82	R	125	}	168	ィ	211	モ	254	
40	(	83	S	126	~	169	ゥ	212	ヤ	255	
41	)	84	T	127		170	ェ	213	ユ		
42	*	85	U	128	♠	171	ォ	214	ヨ		

文字データから文字をとり出す MID$（ミドル・ダラー）

文字データの並べかえは、数値データの並べかえをマスターすれば、案外簡単であることが理解できたと思います。ただ、文字データの場合、ときによっては、個々の文字列のすべての文字を調べて比較するという厳密さを必要としないことがあります。たとえば、最初の1文字だけを調べてアイウエオ順に並べるといったケースです。その方法を考えてみましょう。

この課題を解く手順は、全文字を比較するのとほぼ同じです。違うのは、「入力したデータの文字列から、最初の1文字だけをとり出す」ために、文字列指定関数 **MID$**（ミドル・ダラー）を使うことです。基本形の「式」のところには、数値や数式、あるいは変数を入れることができます。

「式1」は、とり出す文字の最初の位置を指定します。この位置は左からかぞえて何文字めにあたるかで示します。「式2」は、「式1」で指定した位置からとり出す文字数を指定します。「式1」で指定できる範囲は1～255、「式2」は0～255ですが、0を指定すると、空白（ヌルストリング）となります。

「式2」は省略できます。その場合には、「式1」で指定した位置より右にある文字を、すべてとり出します。また、「式2」が「式1」で指定した位置から右にある文字数より多くなった場合も同様です。

文字列指定関数の働きを、次のプログラムで試してみましょう。

基本形

MID$（ミドル・ダラー）**（文字型変数，式1，式2）**

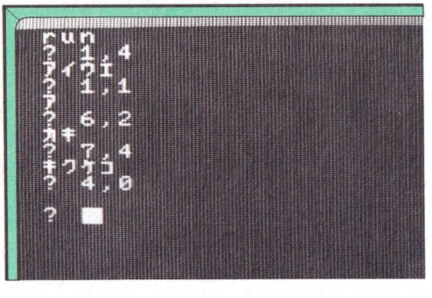
文字列からとり出す文字の位置と長さを指定する文字列指定関数

「式1」と「式2」に数字を与えることによって、「アイウエオカキクケコ」の文字列から自由に文字をとり出すプログラムです。Iが「式1」、Jが「式2」にあたります。

●プログラム

```
10 A$="アイウエオカキクケコ"
20 INPUT I,J
30 PRINT MID$(A$,I,J)
40 GOTO 10
```

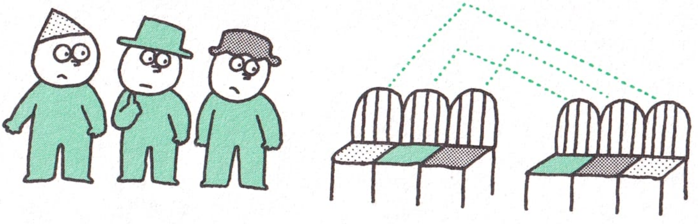
実行



最初の1文字だけの並べかえも可能

これで、入力したデータの最初の1文字をとり出す方法がわかりました。これを比較・並べかえするためには、154ページのプログラムの行100の1部を書きかえて（下のプログラムの行110）、次のプログラムをつくります。

これを実行して、最初の1文字が同じ氏名、最初の2文字が同じ氏名などをデータとして入力してみましょう。このプログラムで同じ順序となるもの、つまり最初の1文字が同じデータは、表示される順序が一定しません。



●プログラム

```
10 N=1
20 DIM D$(10)
30 FOR I=1 TO 10
40    LOCATE 10,I
50    INPUT "ｼﾒｲ";D$(I)
60 NEXT I
70 CLS
80 FOR I=1 TO 9
90    JS=I+1
100   FOR J=JS TO 10
110      IF MID$(D$(I),1,N)>MID$(D$(J),1,N) THEN
         W$=D$(I):D$(I)=D$(J):D$(J)=W$
120   NEXT J
130 NEXT I
140 FOR I=1 TO 10
150    LOCATE 10,I
160    PRINT  D$(I)
170 NEXT I
180 END
```

LESSON—⑤

ひみつの暗号づくり

カタカナが数字に変わる

キャラクタコードを利用すると、おもしろい遊びができます。この章の最後に、暗号づくりと暗号解読のプログラムをつくってみることにしましょう。暗号づくりの手順は、次のようになります。

①画面の文字を消去する。

②ふつうの文を入力する。

③文の１文字をキャラクタコードになおして表示する。

④上記③を文字数だけくり返す。

文字をキャラクタコードになおすには、コード変換関数 **ASC**（アスキー）を使います。基本形のカッコ内の文字は、必ずダブルコーテーション（“ ”）ではさみます。また、「“文字列”」のかわりに文字変数でもかまいません。

基本形

ASC（アスキー）**（“文字”）**

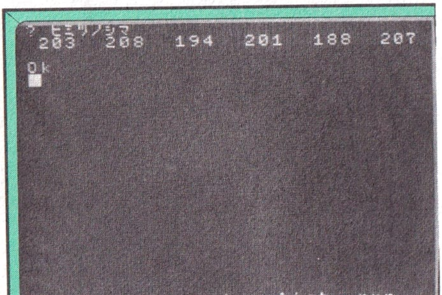
文字を、対応するキャラクタコードの数値に変換するコード変換関数

説明はこのくらいにして、さっそくプログラムを入力してください。実行して、たとえば「ヒミツノシマ」と打ちこんで RETURN キーを押すと、６つの数字に変わります。これが暗号文です。数字は、左から順にそれぞれの文字に対応しているはずです。

●プログラム：暗号づくり

```
10 CLS
20 INPUT A$
30 N=LEN(A$)
40 FOR I=1 TO N
50   AS$=MID$(A$,I,1)
60   PRINT ASC(AS$);
70 NEXT I
```

➔実行

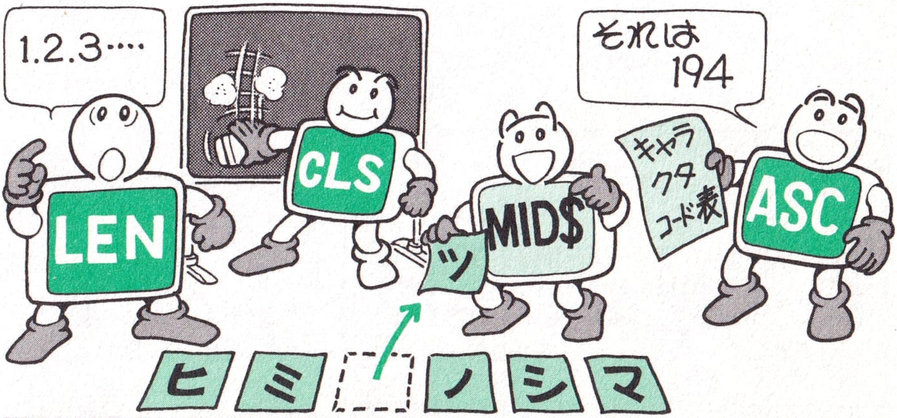


文字の個数をかぞえるLEN(レングス)関数

前ページのプログラムには、新しい命令が2つ出てきました。ひとつは、行10の画面消去命令CLS(クリアスクリーン)です。このおかげで、プログラムを実行するとすぐに画面がきれいになり、入力した文字が見やすくなります。

もうひとつの新しい命令は、文字の数をかぞえる字数関数LEN(レングス)です。行30で使われていますが、なぜ、こんな処理が必要なのでしょうか。

ASC(アスキー)が文字を数字（キャラクタコード）に変える作業は、1文字ずつしかできません。先の「ヒミツノシマ」であれば、変換作業を文字の数だけ、つまり6回くり返します。このくり返しの回数を決めるために、LEN(レングス)によって文字数をかぞえているのです。前ページのプログラムの「N」という変数に注目してください。行30でかぞえた字数の値が代入され、それがそのまま行80でくり返し命令の終値になっています。なお、LEN(レングス)の基本形の「("文字")」は、かわりに文字変数でもかまいません。



基本形

CLS(クリアスクリーン)

画面に表示されている内容をすべて消去する画面消去命令

基本形

LEN(レングス)（"文字"）

文字の数をかぞえる字数関数

暗号解読は CHR$（キャラクタ・ダラー）の役目

前のプログラムでつくった暗号文を、こんどは解読してみましょう。それには、キャラクタコードを、それに対応する文字に変える命令である文字変換関数 **CHR$**（キャラクタ・ダラー）を使います。「式」とは変数（数値型）、数値、または数式のことです。たとえば、「CHR$（177）」はア、「CHR$（200）」はネということになります。グラフィックキャラクタコードの文字や記号は、「CHR$(1)+CHR$(コード+64)」という形で変換します。

暗号解読のプログラムが暗号づくりのプログラムといちばん違うのは、この **CHR$**（キャラクタ・ダラー）と **ASC**（アスキー）の違いですが、そのほかにも注意しなければならないことがあります。ひとつは、くり返しの回数です。暗号文はたいてい3ケタの数字を並べて書いてあり、しかも、それぞれには1字ぶんの空白があります。つまり、カタカナ1文字ぶんは、「空白+3ケタの数字」という形になっていて4文字ぶんあるのです。ですから、暗号文の文字数（空白も1字ぶんとかぞえる）を4で割った数がもとの文章の文字数、つまりくり返しの回数になります。

もうひとつは、文字と数値の型の違いです。数字をひとかたまりずつ（3ケタずつ）とり出すのは **MID$**（ミドル・ダラー）の役目ですが、この関数でとり出された数値は、文字として扱われています。ところが、**CHR$**（キャラクタ・ダラー）のカッコ内には、数値型のものしか入れることができません。そこで、文字（数字）を数値として扱う **VAL**（バリュー）という数値化関数が必要となります。

●プログラム：暗号解読

```
10 CLS
20 INPUT A$
30 FOR I=1 TO LEN(A$) STEP 4
40    AS$=MID$(A$,I,3)
50    PRINT CHR$(VAL(AS$));
60 NEXT I
70 END
```

基本形

CHR$（式）（キャラクタ・ダラー）

数値を、それに対応したキャラクタコードの文字に変換する文字変換関数

基本形

VAL（"文字列"）（バリュー）

文字として扱われている数字を、数値として扱える形にする数値化関数

ゲームで一服

線画遊び

MSX 2パソコンのグラフィック機能は、とても充実している。色もたくさん使えるし、細かい絵も可能。

「でも、プログラムをつくるのがめんどうで……」という人がいるかもしれない。そんな人のためのプログラムがこれ。ジョイスティックを操作するだけで簡単に、しかも自由に線画ができる。256色も使えるから、色の美しさを楽しみながら、ゲーム感覚で絵を描いてみよう。

このプログラムを実行するには、ジョイスティックが必要だ。もしジョイスティックを使わないなら、行290、320、400の「STICK」または「STRIG」に続くカッコ内の値を0にし、行250の「GOSUB」のあとのコンマをとれば、カーソルをカーソル・キーで動かせるようになり、空白キーがトリガボタンのかわりになる。

プログラムの実行を終わりにするには、CTRL を押しながら、STOP を押す。

実行すると、画面の上2／3が白になる。ここが、いわば画用紙だ。下1／3には256色が表示され、これが絵の具箱になる。

画用紙のまん中あたりには、カーソルが表示される。このカーソルは、ジョイスティックで自由に働かせる。

線の色を選択するには、256色のうちの使いたい色のところにカーソルを合わせて、トリガボタンを押す。すると、選んだ色は白い線で囲まれるから、何色を選んだかがわかる。

線を描くには、カーソルを画用紙の部分に移動させ、描きはじめたいところにカーソルを合わせて、トリガボタンを押す。これで、ペンを紙におろした状態になる。あとはジョイスティックを動かすとカーソルが動き、そのとおりに線が描かれる。

もう一度トリガボタンを押すと、ペンを紙から離した状態になる。このままだと、カーソルをどう動かしても、線は描けない。離れた場所にペンをおろしたり、違う色を選んだりするときには、この操作が必要だ。

●プログラム

```
10 DEFINT A-Z:KEY OFF
20 SCREEN 8,2
30 DEFFNP(X,Y,C)=NOT((X>=0) AND (X<256) AND (Y>
   =0) AND (Y<C))
40 DEFFNX(S)=VAL(MID$("112221000",S+1,1))-1
50 DEFFNY(S)=VAL(MID$("100122210",S+1,1))-1
60 '     ショキショリ
70 BX=0:BY=147:BS=8:X1=128:Y1=74:PA$="UP"
80 '     256ショク ノ ヒョウシ゛
90 LINE(0,0)-(255,211),255,BF
100  FOR I=0 TO 255
110     XP=I MOD 32
```

```
120        XP=BX+XP*BS
130        YP=I ¥   32
140        YP=BY+YP*BS
150        LINE(XP,YP)-STEP(BS,BS),1,B
160        LINE(XP+1,YP+1)-STEP(BS-1,BS-1),I,BF
170  NEXT I
180 '      ﾍﾟﾝ ｻｸｾｲ
190  FOR I=1 TO 8
200        READ P$
210        S$=S$+CHR$(VAL("&H"+P$))
220  NEXT I
230 SPRITE$(0)=S$
240 '      ﾍﾟﾝ ﾉ ｱｹﾞｻｹﾞ
250 ON STRIG GOSUB,400
260 '      ﾍﾟﾝ  ﾋｮｳｼﾞ
270 PUT SPRITE 0,(X1,Y1),9,0
280 C=211:Z=5
290 STRIG(1) ON
300 IF PA$="UP" THEN 320
310 IF Y1>146  THEN GOSUB 440 ELSE C=146:Z=1:
    PSET(X1,Y1),C1
320 S=STICK(1)
330 X=X1:Y=Y1
340 '      ﾍﾟﾝ  ｲﾄﾞｳ
350 IF S=0 THEN GOTO 270
360 X=X+(FNX(S)*Z):Y=Y+(FNY(S)*Z)
370 IF FNP(X,Y,C) THEN BEEP ELSE X1=X:Y1=Y
380 GOTO 270
390 '      ﾍﾟﾝ ﾉ ｼﾞｮｳｹﾞ
400 STRIG(1) OFF
410 IF PA$="UP" THEN PA$="DW" ELSE PA$="UP"
420 RETURN
430 '      ｲﾛ ﾉ ｾﾝﾀｸ
440 V1=INT(X1/8):PLAY "C"
450 V2=INT((Y1-147)/8)*32
460 C1=V1+V2
470 IF SY<>0 THEN LINE(SX,SY)-STEP(BS,BS),0,B
480 SX=(C1 MOD 32)*BS+BX:SY=(C1¥32)*BS+BY
490 LINE(SX,SY)-STEP(BS,BS),255,B
500 PA$="UP"
510 RETURN
520 '      ﾍﾟﾝ ﾉ ﾃﾞｰﾀ
530 DATA FF,FE,FC,FC,FE,FF,CF,86
```

PART 8



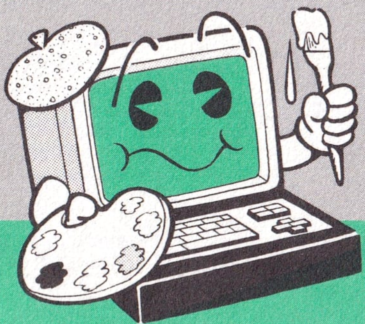


楽しもう色と形

パソコン・グラフィックス

MSX パソコンには、豊富なグラフィック機能がついています。グラフィック画面を使って細かく、きれいなグラフィックスを描くことができます。この機能をフルに活用して、私たちの目を楽しませてくれるプログラムをつくりましょう。

この章で学ぶ命令は、SCREEN、COLOR、LINE、CIRCLE、GOSUB、RETURN、ON〜GOSUB、PAINT、REM、INKEY$ などです。



LESSON ①

まず画面の選択から始めよう

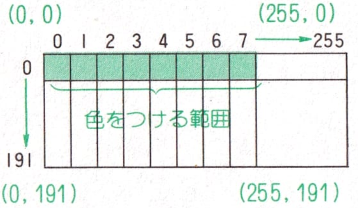
絵を描くにはグラフィック画面

MSX 2 パソコンには、文字だけを書くテキスト画面、絵を描くグラフィック画面(文字も書ける)とがあります。テキスト画面は2種類でしたが、グラフィック画面は大きく分けて、高解像度グラフィック、低解像度グラフィック、ビットマップグラフィックと3種類あります。それぞれの画面になる状態のことをモードといい、高解像度グラフィック・モード、低解像度グラフィック・モードなどといいます。それぞれの特徴は次のとおりです。

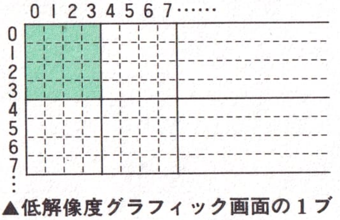
●高解像度グラフィック・モード　横256個×縦192個の点で絵を描く。色は512色中16色使え、ドット単位で色をつけられますが、横8×縦1の点のブロックごとに同時に2色までしか使えません。同じブロック内で3番めの色をぬると、2番めにぬった色が、その色に変化してしまうので注意しましょう。

●低解像度グラフィック・モード　横64×縦48のブロックに分かれていて、各ブロックごとに色をつけられます(512色中16色)が、1ブロックが横4×縦4の点でできているので、細かい絵は描けません。

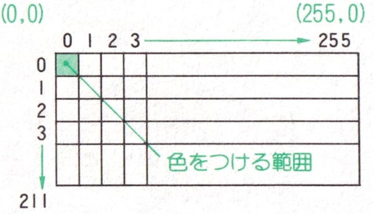
●ビットマップグラフィック・モード　横256個(または512個)×縦212個の点で絵を描きます。1個1個の点ごとに512色中16色(SCREEN 8なら256色)を使って色をつけることができるので、細かい絵を描くのに適しています。このモードは、細かく分けると4種類あります。



▲高解像度グラフィック画面



▲低解像度グラフィック画面の1ブロック



▲ビットマップグラフィック画面(SCREEN 5〜8の場合)

画面を設定するには SCREEN 命令

MSX 2パソコンのグラフィック画面には、下にあげたように7種類のモードがあります。文字を表示するテキスト画面については、すでに64ページでお話ししたように2種類ありますから、合計で9種類のモードがあることになります。

このうち、どのモードを選ぶかは、モード設定命令 **SCREEN** を使います。基本形の数値は0～8の範囲で、この数によってモードが決まります。ただし、「SCREEN　7と8」のモードは、VRAMが128KBの機種でしか使えません。注意してください。

基本形

SCREEN␣数値

画面のモードを決めるモード設定命令

なお、テキスト・モードとグラフィック・モードは、同時に使うことはできません。このため、命令によっては、どちらか一方のモードでしか使えないものもあります。

◉モードの種類と特徴

モード		指定の方法	解像度	同時に使える色	備考
テキスト	I	SCREEN 0		512色中2色	64ページ参照
テキスト	II	SCREEN 1		512色中16色	64ページ参照
グラフィック	高解像度	SCREEN 2 SCREEN 4	256×192	512色中16色	色は横8点ごとに2色まで
グラフィック	低解像度	SCREEN 3	64×48	512色中16色	ブロックごとに着色
グラフィック	ビットマップ	SCREEN 5	256×212	512色中16色	多くの色が使える
グラフィック	ビットマップ	SCREEN 6	512×212	512色中4色	細かい絵が描ける
グラフィック	ビットマップ	SCREEN 7	512×212	512色中16色	VRAMが128KBの機種でだけ指定できる
グラフィック	ビットマップ	SCREEN 8	256×212	256色	VRAMが128KBの機種でだけ指定できる

絵を描き続けるには無限ループで

MSX 2 パソコンで絵を描く際に注意したいのは、

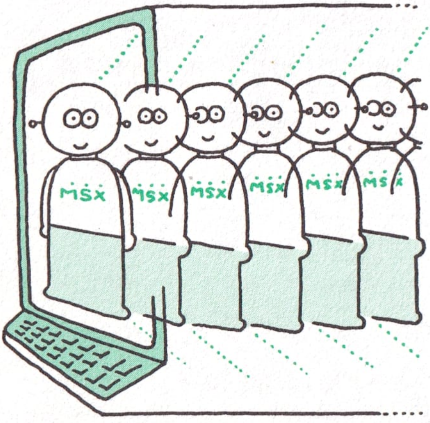
①プログラムの実行が終わったとき

②**INPUT**(インプット) 文や **PRINT**(プリント) 文が実行されたとき

には、必ずテキスト・モードに戻る、ということです。なかでも注意したいのは①です。絵を描くプログラムを実行した場合、たしかに絵は描けますが、一瞬で消えてしまいます。テキスト・モードに戻って、描かれていた絵は消え、カーソルと「Ok」表示が出てきます。

絵をいつまでも表示させておくには、特別の処理をしてグラフィック・モードが続いているようにします。たとえば、プログラムの最後にゆく先指定命令 **GOTO**(ゴーツー) を加え、ゆく先には **GOTO**(ゴーツー) 文のある行を指定します。これなら、プログラムは無限ループになって、実行が終わらないので、絵を描き続けることができます。下の例ではこの方法をとっています。

下のプログラムで、行 50 を **END**(エンド) にした場合はどうなるか、修正して実行し、結果の違いを比べてみてください。



無限ループにした場合の画面は、見た目には止まっているように見えますが、実際には同じプログラムがくり返し実行されているのです。止まっているように見えるのは、実行速度がきわめて高速のためです（81、82 ページ参照）。

●プログラム

```
10 SCREEN 5
20 COLOR 1,1,1
30 CLS
40 LINE (50,70)-(150,110),10
50 GOTO 50
```

絵の色、背景の色を自由に指定

画面の色は、色指定命令 **COLOR**（カラー）で指定することができます。基本形の「COLOR」のあとの①、②、③は数値で、それぞれ次のことを示しています。

①——表示色（フォアグラウンドカラー）といい、テキスト画面の文字や、グラフィック画面の点や線などの色として使う。ほかの方法で色を指定しなければ、この色で表示される。

②——背景色（バックグラウンドカラー）といい、画面の地の色として使う。

③——周辺色（ボーダーカラー）といい、画面の周辺部の色。この部分は、ベーシックでは使えない。

①〜③は、パレット番号で指定します。絵を描くときに使うパレットには、白いマス目がいくつか並んでいます。MSX 2 パソコンにもそのようなマス目が16個あり、0〜15の番号がついていると考えてください（「SCREEN 6」の場合は、マス目が4個で番号は0〜3）。電源を入れたときの画面は、自動的に「COLOR 15, 4, 7」になります。

各パレット番号の色は、電源を入れたあとには、別表のようになります。ですから、電源を入れたすぐあとの画面は、表示色が白、背景色が暗い青、周辺色が水色です。

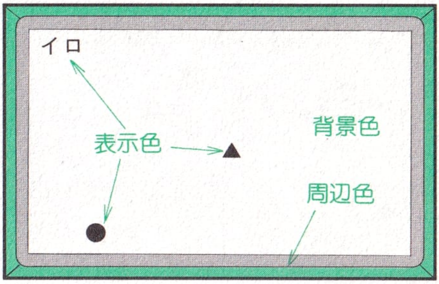
◉電源投入後のパレット

パレット番号	色	パレット番号	色
0	透明	8	赤
1	黒	9	明るい赤
2	緑	10	黄
3	明るい緑	11	明るい黄
4	暗い青	12	暗い緑
5	明るい青	13	紫
6	暗い赤	14	灰色
7	水色	15	白

基本形

COLOR（カラー）␣①，②，③

グラフィック画面の表示色、背景色、周辺色を指定する色指定命令



LESSON ②

命令ひとつで四角も円も

直線を描くにはLINE命令

いろいろな図形や絵を描くための命令を紹介します。まず、直線を描く直線命令 **LINE** があります。基本形の座標①と座標②を指定すると、この２点を結んだ直線になります。パレット番号を指定すると、その色の線になりますが、省略すると色指定命令 **COLOR** で指定した表示色が採用されます。パレット番号の後ろにB（Boxの頭文字）を入れると、上の２点を対角線に持つ四角形を描きます。BF（Box Fillの頭文字）を入れると、その四角形をぬりつぶします（後述参照）。

次のような手順で座標（50, 70）、（150, 110）の２点を結ぶ線を描くプログラムをつくります。

①モードを設定する（**SCREEN**）。

②直線を指定どおりに描く（**LINE**）。

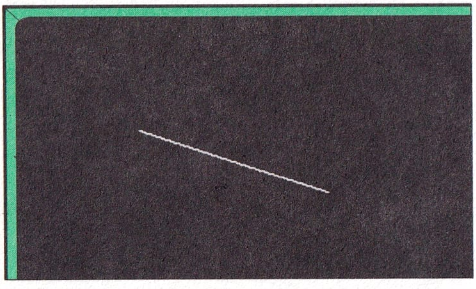
③無限ループにする（**GOTO**）。

モードは、「SCREEN 5」のモードにします。

●プログラム：直線を描く

```
10 SCREEN 5
20 COLOR 1,1,1
30 CLS
40 LINE (50,70)-(150,110),10
50 GOTO 50
```

実行



基本形

LINE␣(座標①)−(座標②), カラーコード, B(またはBF)

画面上の指定した２点間に直線を引く直線命令

いろいろな色で線を描く

直線命令を使って、でたらめな線を10本描くごとに、色が次々に変わるプログラムをつくってみましょう。手順は次のようになります。

①モードを設定する（「SCREEN␣5」のモード）。

②でたらめな座標を設定する（横0～255、縦0～211の範囲で **RND**（ランダム）と **INT**（インテジャー）を使う）。

③上記②の座標で直線を描く。

④上記②と③を10回くり返して10本の線を描く（**FOR～NEXT**（フォー・ネクスト））。

⑤線の色を変える。

⑥上記②～⑤をくり返す。

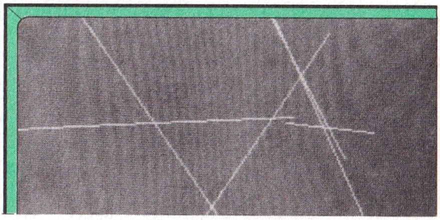
問題は⑤です。色を次々に変えるのですから、**LINE**（ライン）文の色指定の部分を変数として、その値を **FOR～NEXT**（フォー・ネクスト）文で変化させればよいでしょう。

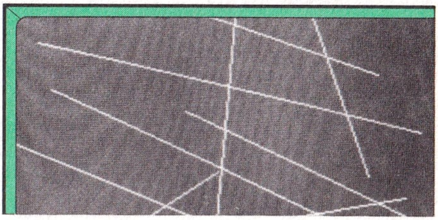
このプログラムを実行すると、でたらめに10本の直線が描かれ、そのあとで直線の色が変わっていきます。Iの値は0から15まで連続的に変わるので（行20～90）、169ページ別表の順番に色が変わります。

●プログラム：10本の線を描き、その色を変える

```
10 SCREEN 5
20 FOR I=0 TO 15
30   CLS
40   FOR J=1 TO 10
50     X1=INT(RND(1)*256):Y1=INT(RND(1)*212)
60     X2=INT(RND(1)*256):Y2=INT(RND(1)*212)
70     LINE (X1,Y1)-(X2,Y2),I
80   NEXT J
90 NEXT I
100 GOTO 20
```

実行





背景の色、線の色は自分で決めよう

表示色と背景色を、自分で決められると便利です。どんな色を組み合わせればどんな感じになるか、配色のぐあいを見ることができます。

このプログラムでは、モード設定命令**SCREEN**（スクリーン）の位置に注意してください。表示色や背景色は、数値を入力することによって選ぶわけですが、プログラムの最初でグラフィック・モードを指定しても、**INPUT**（インプット）文や**PRINT**（プリント）文が実行されたとたんにテキスト・モードに戻ってしまいます。これでは、絵が描けません。

ですから、モード指定は、**INPUT**（インプット）文や**PRINT**（プリント）文を実行したあとで行なうことにします。また、描いた線をそのまま残しておくために、無限ループにする必要があります。この場合は、プログラムの最初の行に戻るのではなくて、同じ行でループする形にします。

実行すると、色と数値の組み合わせが表示されます。同じ色の名前では「−1」となっているのが明るい色、「−2」となっているのが暗い色です。たとえば、「ミドリ」はふつうの緑、「ミドリ−1」は明るい緑、「ミドリ−2」は暗い緑です。色を適当に選んで入力すると、画面の文字が消えて、すぐに背景が指定の色になり、指定した色の線が10本描かれます。実行を中断するときには、CTRL キーを押しながら STOP キーを押します。

●プログラム：背景の色、線の色を決める

```
10 PRINT "1=ｸﾛ,2=ﾐﾄﾞﾘ,3=ﾐﾄﾞﾘ-1,4=ｱｵ-2,5=ｱｵ-1,
   6=ｱｶ-2,7=ﾐｽﾞｲﾛ,8=ｱｶ,9=ｱｶ-1,10=ｷｲﾛ,11=ｷｲﾛ-1,
   12=ﾐﾄﾞﾘ-2,13=ﾑﾗｻｷ,14=ｸﾞﾚｰ,15=ｼﾛ"
20 INPUT "ﾊﾞｯｸ ﾊ ﾅﾆｲﾛ? (1-15)";B
30 INPUT "ｾﾝ ﾊ ﾅﾆｲﾛ? (1-15)";L
40 SCREEN 5
50 COLOR B,B,B
60 CLS
70 FOR I=1 TO 10
80   X1=INT(RND(1)*256):Y1=INT(RND(1)*212)
90   X2=INT(RND(1)*256):Y2=INT(RND(1)*212)
100   LINE (X1,Y1)-(X2,Y2),L
110 NEXT I
120 GOTO 120
130 END
```

四角形は2点を指定すれば描ける

これから、グラフィック命令を使って、さまざまな図形や絵を描きます。図を描くための命令の説明を中心にするので、これ以降のプログラムでは、画面のモードをビットマップグラフィック・モードのひとつ「SCREEN 5」に統一することにします。

まず、四角形を描いてみましょう。すでに説明したように四角形は直線命令の**LINE**を使って描きます。たとえば、グラフィック座標の（120, 40）と（200, 120）を結ぶ線を対角線とする四角形を描く命令は、次のようになります。

LINE　(120, 40)－(200, 120), 10, B

線の色を黄（パレット番号）にする ─↑　↑四角形にする

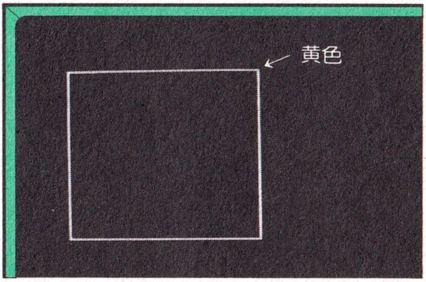
このように、対角線をつくる2点を決めれば、四角形の位置と大きさが決まります。

四角形の線の色は、黄色にしています。

この命令を中心にして、きちんとしたプログラムにすると、下のようになります。単純に線を引くだけでも、これだけのプログラムになるのです。

このプログラムと169ページのプログラムの実行結果を比べてみてください。「,B」がつくだけでこれだけ違う結果が出ます。また、行40の最後の「B」を「BF」にすると、四角全体が黄色でぬりつぶされます。試してみましょう。

実行



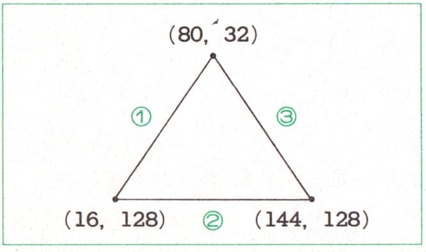
●プログラム：黒地に黄色の四角形を描く

```
10 SCREEN 5
20 COLOR 1,1,1
30 CLS
40 LINE (120,40)-(200,120),10,B
50 GOTO 50
```

直線命令で三角形を描く

直線命令を使って、三角形を描いてみましょう。三角形は3点を結ぶ直線ですから、3点の座標をとって、それぞれに直線命令を実行すれば描けます。

たとえば、(80, 32)、(16, 128)、(144, 128) の3点を結ぶ三角形を描くとすると、図のように①、②、③と3本の直線を引くことになります。3点の座標のとり方によって、正三角形、直角三角形、二等辺三角形なども自由に描けます。表示する位置や形を正確に決めるためには、256×212のグラフを別につくり、その上で図形を描いて座標を決めてから、プログラムをつくるようにします。

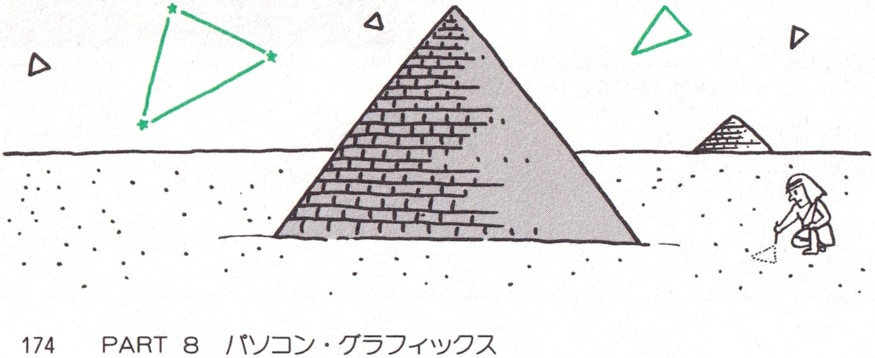


▲3点を結ぶ直線が3本で三角形

直線命令では、左側の座標指定をしないと、直前に描かれた直線の終点（直線命令でいえば右側の座標）が、新しい直線の始点になります。この方法を利用すると、行50の左側(16, 128)、行60の左側(144, 128)は必要なくなり、プログラムが簡単になります。

●プログラム：三角形を描く

```
10 SCREEN 5
20 COLOR 1,1,1
30 CLS
40 LINE (80,32)-(16,128),10
50 LINE (16,128)-(144,128),10
60 LINE (144,128)-(80,32),10
70 GOTO 70
```



横にずれる三角形

前出のプログラムに少し手を加えて、三角形が横に移動し、そのたびに色が変わるようにしてみましょう。三角形を横に動かすためには、3点の水平座標を移動させます。三角形の色を変えるには、**LINE**文のパレット番号を変えなくてはなりません。背景色が黒ですから、表示色は黒以外の色にします。

加える手順は、次のとおりです。

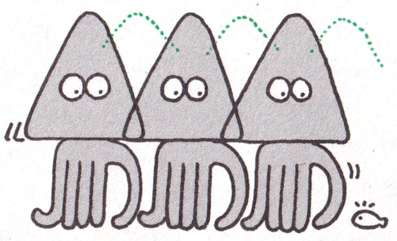
①表示色を変える。

② 3点の座標を横に移動する。

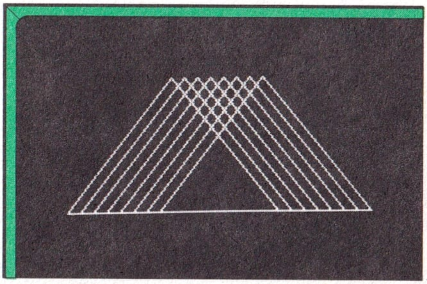
③上記の①と②をくり返す。

ここでは、行30～100のくり返し命令**FOR～NEXT**で三角形の色を変えます。

行50の「STEP　8」は、横にずれる幅です。座標で8ずつ右へ動きます。動き方をもっと小さくするなら、STEPのあとの数字を8より小さい数にします。逆に、動き方を大きくするには、この数字を8より大きい数にします。



実行



●プログラム：三角形が横にずれて、色が変わる

```
10 SCREEN 5
20 COLOR 1,1,1
30 FOR I=2 TO 15
40   CLS
50   FOR J=8 TO 64 STEP 8
60     LINE (80+J,57)-(16+J,153),I
70     LINE (16+J,153)-(144+J,153),I
80     LINE (144+J,153)-(80+J,57),I
90   NEXT J
100 NEXT I
110 GOTO 110
```

半径を変えれば、同心円も簡単に

円を描くには、新しい命令が必要です。基本形で、中心の座標の設定方法は、これまでと同じです。半径は、座標の長さで指定します。

パレット番号は、その円をどの色で描くかを指定するものです。これを省略すると、色指定命令 **COLOR**（カラー）で指定した表示色の円になります。

開始角度および終了角度は、円弧を描く場合に指定します。このどちらかを負の値にすると、角度はその絶対値となり、中心からの半径が描かれます。両方とも負の値にすると、扇形になります。角度は、「$-2\pi \sim +2\pi$」の範囲内にします。これを超えると、「Illegal function call」（イリーガル ファンクション コール）というエラーメッセージが出ます。円弧を描くとき以外は、開始角度、終了角度とも省略します。

比率は、円またはどの程度のだ円にするかを指定するもので、「垂直方向の半径÷水平方向の半径」として入力します。円を描く場合には、比率を1とします。省略すると、1と解釈され、自動的に正円になります。

座標（128, 106）を中心とし、半径を変数とするプログラムは、下のようになります。これを実行すると、同心円がいくつも描かれます。また、行50でパレット番号（変数I）を2～15に変えてありますから、描かれる円は、すべて色が変わります。同じ行で半径を「I＊6」としてあるのは、すべての円を画面におさめるための処置です（最大直径180）。

たんに円を1つ描くだけなら、プログラムの行40以下を、次のようにすればよいのです（半径は48とする）。

```
40 CIRCLE (128,
   106), 48, 3
50 GOTO 50
```

基本形

CIRCLE（サークル）␣**(中心の座標), 半径, パレット番号, 開始角度, 終了角度, 比率**

円またはだ円を描く円命令

●プログラム：同心円を次々に別の色で描く

```
10 SCREEN 5
20 COLOR 1,1,1
30 CLS
40 FOR I=2 TO 15
50   CIRCLE (128,106),I*6,I
60 NEXT I
70 GOTO 70
```

円命令でだ円を描くには

だ円を描くには、円命令の比率を1以外の値にします。比率が1を超えると、垂直方向の半径を基準にして縦に長いだ円になり、1未満なら、水平方向の半径を基準にして横に長いだ円になります。

中心が(128,106)で、水平方向の半径が56、比率0.7のだ円を描くには、次のようにします。最後の「.7」が比率で、このように、小数点の前につくゼロは省略することができます。

```
CIRCLE (128,106), 56, 10, , , .7
```

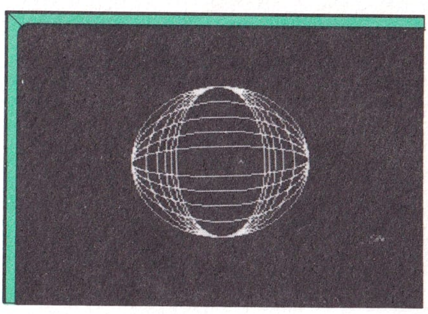
開始角度と終了角度を省略し、比率を書く場合には、省略した2つの角度に付随するコンマ(,)が省略できません。ただし、比率まで省略する場合には、コンマは不要です(前ページのプログラム行50を参照)。

比率を変数にして、同心状のだ円をいくつも描くプログラムは、下のようになります。

●プログラム:同心状のだ円をいくつも描く

```
10 SCREEN 5
20 COLOR 1,1,1
30 CLS
40 FOR I=.2 TO 2 STEP .2
50   CIRCLE (128,106),56,10,,,I
60 NEXT I
70 GOTO 70
```

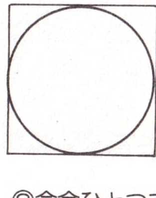
実行



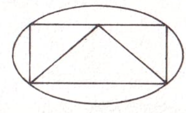
練習問題12

次の図のような図形を描くプログラムを作成してみよう。画面上の図の位置や大きさ、色は自由。

①



②



②命令ひとつで四角も円も

LESSON ③

いろんな図形を自由に指定

決まった仕事はサブルーチンで実行

プログラムの中に、三角形、四角形、円、だ円をセットしておいて、指定した数値によってどれか1種類を選べる、という課題を解いてみましょう。図形を描く部分は、いわば決まりきった仕事ですから、プログラムの中で小さなブロックにまとめ、必要に応じ、数値によって呼びだすようにすると便利です。このブロックを「サブルーチン」といいます。

サブルーチンを呼びだして、その実行を行なうには、サブルーチン呼びだし命令 **GOSUB**（ゴーサブ）を使います。また、サブルーチンの最後には帰還命令 **RETURN**（リターン）を置きます。この2つの命令は、必ず対で使われますから、**GOSUB〜RETURN**（ゴーサブ・リターン）で1つの命令として考えてください。

●プログラム：直方体の体積と表面積を求める

```
10 INPUT "ﾀﾃ=";A
20 INPUT "ﾖｺ=";B
30 INPUT "ﾀｶｻ=";H
40 GOSUB 90
50 PRINT "ﾀｲｾｷ=";S
60 PRINT "ﾋｮｳﾒﾝｾｷ=;S
70 PRINT
80 GOTO 10
90 V=A*B*H
100 S=2*(A*B+A*H+B*H)
110 RETURN
```

基本形

GOSUB（ゴーサブ）␣行番号
　〜
RETURN（リターン）

サブルーチンを呼びだし、その処理が終わるともとのプログラムへ帰還する、サブルーチン呼びだし命令

GOSUB（ゴーサブ）に続く行番号は、サブルーチンのはじめの行番号です。これにより、サブルーチンの処理が行番号順に行なわれ、**RETURN**（リターン）にくると、**GOSUB**（ゴーサブ）文のすぐあとの命令文に戻ります。**GOSUB**（ゴーサブ）文がないのに **RETURN**（リターン）に出会うと、「RETURN（リターン） without（ウィズアウト） GOSUB（ゴーサブ）」というエラーメッセージが出ます。

たとえば、例にあげた、直方体の体積と表面積を求めて表示するプログラムでは、計算する部分（行90〜100）をサブルーチンとしています。行40でサブルーチンを呼びだし、行90と行100を実行して、行110で「RETURN」にぶつかると、行50に帰還します。

4つの図形を数字で呼びだす

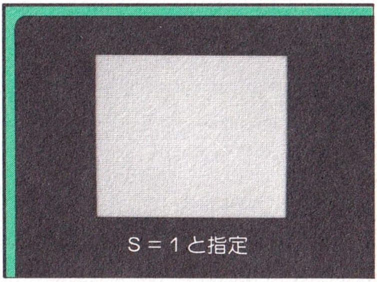
ここでの課題のように、サブルーチンがいくつもあって、値によってどのサブルーチンを実行するかを指定するときには、サブルーチン指定命令 **ON〜GOSUB**（オン・ゴーサブ）を使います。基本形のうち、「数式」は変数でもかまいませんが、その値（プログラム行60のS）は1から始め、行番号の並び順によって、下のプログラム例に矢印で示したように、それぞれの値に対応します。

基本形

ON（オン）␣数式GOSUB（ゴーサブ）␣行番号，行番号，行番号……

指定した条件を満たしたときに、サブルーチンを呼び出して実行するサブルーチン指定命令

実行



●プログラム：４種類の図形を指定する

```
10 COLOR 15,4,7
20 INPUT "ｼｶｸ=1,ｻﾝｶｸ=2,ｴﾝ=3,ﾀﾞｴﾝ=4";S
30 SCREEN 5
40 COLOR 1,1,1
50 CLS
60 ON S GOSUB 80,100,150,180
70 FOR I=1 TO 800:NEXT I:GOTO 10
80 LINE (88,65)-(168,145),10,BF
90 RETURN
100 LINE (128,57)-(64,153),10
110 LINE (64,153)-(192,153),10
120 LINE (192,153)-(128,57),10
130 PAINT (128,58),10
140 RETURN
150 CIRCLE (128,105),48,10
160 PAINT (128,105),10
170 RETURN
180 CIRCLE (128,109),56,10,,,.7
190 PAINT (128,109),10
200 RETURN
```

S=1（→80）　S=2（→100）　S=3（→150）　S=4（→180）　帰還（RETURN→70）

1点指定で図形を彩色

前出のプログラムでは、行130などに新しい命令が使われています。色をぬるために使う彩色命令 **PAINT**（ペイント）です。基本形のうち、ぬりはじめの座標は、ぬりたい図形の内側であれば、どこでもかまいません。ただ、細かくいえば、ぬられていく過程が違います。図形の上のほうに座標をとれば、上から下へ、まんなかに座標をとれば、上下に向けて同時に色がぬられていきます。

ぬる色と境界色は、パレット番号（169ページ参照）で指定します。ぬる色の指定を省略すると、色指定命令 **COLOR**（カラー）で指定された表示色と同じ色になります。

基本形

PAINT␣（ぬりはじめの座標），ぬる色，境界色

指定した境界色で囲まれた部分を、指定した色でぬる彩色命令

また、境界色が指定できるのは、モードのうち「SCREEN　3と5～8」だけです。「SCREEN　2と4」では、ぬりつぶす色がそのまま境界色になります。

では、彩色命令の指定によって実行結果がどう違うか、本項のプログラムで調べてみましょう。実行すると、Sの値によってぬる順序の変わることがわかります。

S=1とすると、色がまんなかからぬられていきます（行100）。S=2では、色が上から下へとぬられます（行120）。S=3の場合は、下から上へとぬられます（行140）。S=4の場合は、円の外側にぬりはじめの座標を指定したため、円の外側が指定した色でぬりつぶされます（行160）。

図形の線が切れていると、内側に指定しても外側までぬられるので注意しましょう。

●プログラム：円をぬりつぶす

```
10 INPUT "1-4";S
20 SCREEN 5
30 COLOR 1,1,1
40 CLS
50 CIRCLE (128,106),64,10
60 ON S GOSUB 100,120,140,160
70 FOR I=1 TO 800:NEXT I
80 COLOR 15,4,7
90 GOTO 10
100 PAINT (128,106),10
110 RETURN
120 PAINT (128,43),10
130 RETURN
140 PAINT (128,169),10
150 RETURN
160 PAINT (194,106),10
170 RETURN
```

パレット番号の色を変えるには

これまで、いろいろなプログラムで色を指定するのに使ったパレット番号は、各番号によって色が一定でした。たとえば、パレット番号が2なら、必ず緑になります。番号と色が、いつも、169ページ別表のとおりに対応しているからです。この状態を「パレットの初期状態」といいます。

前にお話ししたようにMSX 2パソコンでは512色が使えるはずなのに、これでは、けっきょく16色しか使えません。表にある16色以外の色を使うためには、パレット番号に割りあてられている色を変える必要があります。それには、色指定命令 **COLOR**（カラー）を使います。この命令には、これまで説明した(169ページ)ほかに、パレットに好みの色を割りあてる働きがあります。

基本形
COLOR＝(パレット番号,①,②,③)
パレット番号の色を指定する

基本形の①、②、③は、それぞれ赤、緑、青の濃さをあらわす数値で、0～7の値で指定します。たとえば「赤7、緑7、青0」とすると、明るい黄色になります。

では、試してみましょう。次のようにキーインしてください。これで、パレット番号の10に、「赤2、緑3、青4」の色が割りあてられたはずです。

```
COLOR=(10, 2, 3, 4)  RETURN
```

ついで、前ページのプログラムを実行すると、前に実行したときとは違ってグレーの円になります。①、②、③の数をいろいろ変えて、どんなふうに色が変わっていくかを調べましょう。

「COLOR＝」命令文で、パレット番号の色を変えると、プログラムの実行後も、その状態が続くので要注意です。そのまま、別のプログラムを入力して実行すると、出そうと思った色とは違う色が出ることがあります。

パレットを初期状態に戻すには、一度電源を切るか、下の命令を実行します。プログラム中なら色指定を行なう命令の前にこの命令を置きます。プログラムを入力する前に、この命令文をキーインして RETURN キーを押しても、同じことです。

```
COLOR=NEW
```

パレットの色を一瞬に変える

パレットに色を割り当てる働きを応用すると、すでに描いた図形の色を一瞬にして変えることができます。

人気者のパンダの顔を画面に描き、白と黒の色の組み合わせを次々に変えていくプログラムをつくってみましょう。パンダの顔は、円命令 **CIRCLE** を使ってつくった円を組み合わせれば描けます。耳と目と鼻が黒、そのほかは白ですが、色の組み合わせを変えるためには、耳や目、鼻などの黒い部分のパレット（番号は 1）と、顔や目玉などの部分の白い部分のパレット（番号は15）に割りあてる色を変えることになります。そのためには、次のようにします。

①「**COLOR** =」命令文の赤、緑、青を変数にする。

ここではPR、PG、PBという変数を、パレット1とパレット15の両方に使っています。パレット1では、赤の濃さをPR、緑をPG、青をPBに、パレット15では赤をPB、緑をPR、青をPGとしています。

②上記①の変数を **FOR** ～ **NEXT** で変える。

ここでは、行30～140のループでPR、PG、PB 3つの変数の値を変えています。行30～140の部分では、ループが三重になっていることに注目してください。J、K、Lの数を変え、それぞれを8で割った余りをPR、PG、PBの値としています。単純なやり方としては、どの変数も0→7と変えればよいのですが、これでは、つねに、色の薄いほうから濃いほうへの変化だけで、おもしろみがないので、上のようなめんどうな手続きを踏んだわけです。

このプログラムの実行を終了するには、CTRL キーを押しながら STOP キーを押します。

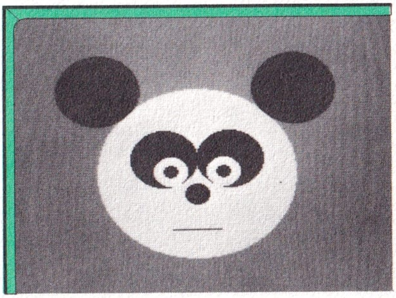
●プログラム：パンダの顔の色を変える

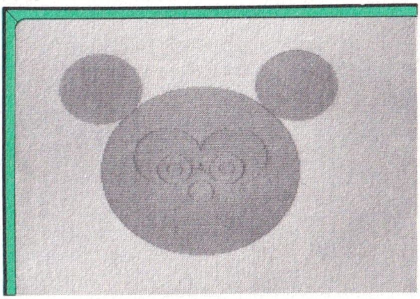
```
10 COLOR 15,4,7:SCREEN 5
20 GOSUB 170
30 FOR J=0 TO 7
40  FOR K=J+1 TO J+8
50   FOR L=K+1 TO K+8
60    PR=J MOD 8
70    PG=K MOD 8
80    PB=L MOD 8
```

```
90    COLOR=(1,PR,PG,PB)
100     COLOR=(15,PB,PR,PG)
110     FOR M=1 TO 100:NEXT M
120    NEXT L
130   NEXT K
140 NEXT J
150 GOTO 10
160 'ﾊﾟﾝﾀﾞ ｦ ｴｶﾞｸ
170 CIRCLE (120,120),60,15
180 PAINT (120,120),15
190 CIRCLE (100,110),20,1
200 PAINT (100,110),1
210 CIRCLE (140,110),20,1
220 PAINT (140,110),1
230 CIRCLE (60,60),25,1
240 PAINT (60,60),1
250 CIRCLE (180,60),25,1
260 PAINT (180,60),1
270 CIRCLE (104,119),10,15
280 PAINT (104,119),15
290 CIRCLE (104,119),4,1
300 PAINT (104,119),1
310 CIRCLE (136,119),10,15
320 PAINT (136,119),15
330 CIRCLE (136,119),4,1
340 PAINT (136,119),1
350 LINE (105,160)-(134,160),1
360 CIRCLE (120,135),7,1
370 PAINT (123,135),1
380 FOR I=1 TO 500 :NEXT I
390 RETURN
```

60.12.22

実行





512色をすべて表示すると

下のプログラムを実行すると、色が12色ずつあらわれ、512色のすべてを見ることができます。じっくり見るには、STOP キーを押して画面を停止させます。もう一度押せば、再び次々に色が変わります。各色の下にある3ケタの数字は、その色の赤、緑、青の濃さです。181ページの基本形にあてはめてみてください。

●プログラム：512色を12色ずつ表示する（実行結果はカラー口絵）

```
10 OPEN "GRP:" FOR OUTPUT AS #1
20 DEFINT A-Z
30 BX=35:BY=35:BS=48
40 COLOR 15,1,1:SCREEN 5
50 PSET (81,25),POINT (81,25)
60 PRINT #1,USING "@";"COLOR=(R,G,B)"
70 FOR R=0 TO 7
80  FOR G=0 TO 7
90   FOR B=0 TO 7
100    XC=A MOD 12
110    XP=XC MOD 4
120    XP=BX+XP*BS
130    YP=XC ¥   4
140    YP=BY+YP*BS
150    XC=XC+3
160    COLOR=(XC,R,G,B)
170    LINE(XP,YP)-STEP(BS,BS),XC,BF
180    S$="("+RIGHT$(STR$(R),1)+","+RIGHT$(STR$
       (G),1)+","+RIGHT$(STR$(B),1)+")"
190    X=XP+5:Y=YP+23:BC=5
200    FOR L=1 TO LEN(S$)
210     PSET (X,Y),POINT (X,Y)
220     PRINT #1,MID$(S$,L,1)
230     X=X+BC
240    NEXT L
250    FOR J=1 TO 200:NEXT J
260    A=A+1
270   NEXT B
280  NEXT G
290 NEXT R
300 GOTO 300
```

「SCREEN　8」で使える256色

これまでのプログラムでは、色を指定するのにすべてパレット番号を使ってきました。このパレット番号には、全部で512色を割りあてることができましたが、実際に1つのプログラムで使える色は最高16色でした。ところが、「SCREEN　8」のモードでは、同時に256色が使えます。

参考のために、256色のそれぞれがどんな色かがわかるプログラムをつくりました。実行すると、画面には256色のすべてが一度に表示されます(口絵参照)。ずいぶんたくさんの色を使えることがわかるでしょう。

「SCREEN　8」で色を指定するには、パレット番号ではなく、それぞれの色について決まっている番号(カラーコード)を使います。このプログラムでは、その番号を16進数で各色の下に表示しています。色を指定するには、「&H」のあとにこの番号を書きます。たとえばF0という番号の色を指定するには、「&HF0」と書けばよいのです。

なお、&Hは、あとに続く数値が16進数であることを示す記号です。

●**プログラム：256色を一度に表示する**(実行結果はカラー口絵)

```
10 A=8:B=2:C=1
20 OPEN "GRP:" FOR OUTPUT AS #1
30 BX=27:BY=1:BS=13
40 SCREEN 8:COLOR 255
50 FOR I=0 TO 255
60   XP=I MOD 16
70   XP=BX+XP*BS
80   YP=I ¥  16
90   YP=BY+YP*BS
100  LINE(XP,YP)-STEP(BS,BS),,B
110  LINE(XP+1,YP+1)-STEP(BS-2,BS-2),I,BF
120  PSET (XP+B,YP+C),POINT (XP+B,YP+C)
130  D$=RIGHT$(SPACE$(1)+HEX$(I),2)
140  PRINT #1,LEFT$(D$,1)
150  PSET (XP+A,YP+C),POINT (XP+A,YP+C)
160  PRINT #1,RIGHT$(D$,1)
170 NEXT I
180 GOTO 180
```

LESSON ④

画面いっぱいにカラフルな図形を

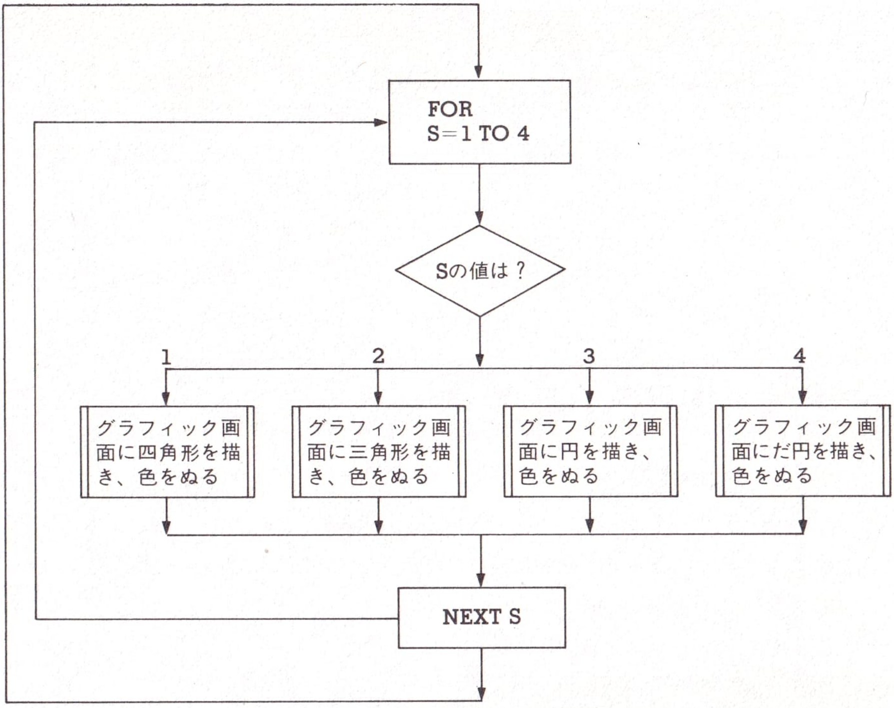
無限ループで４種の図形を次々に

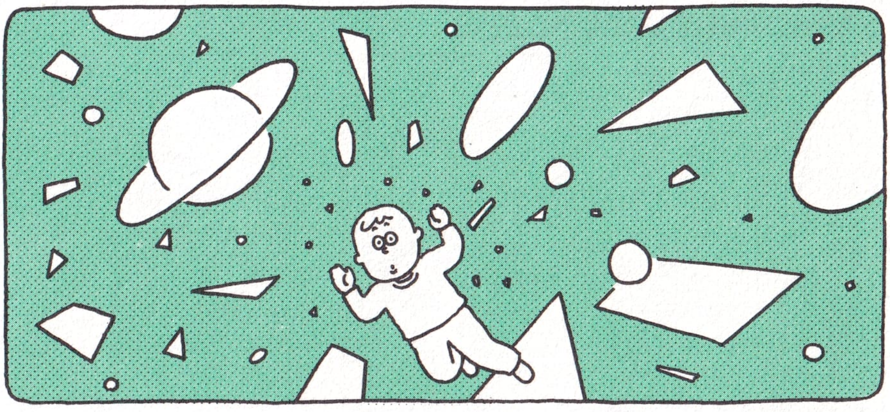
サブルーチン指定命令 **ON～GOSUB**（オン・ゴーサブ）をくり返し命令 **FOR～NEXT** と組み合わせて使うと、いくつものサブルーチンを次々に実行するプログラムができます。

前に使った図形を選ぶプログラムを手なおしして、４種類の図形が次々に出てくるプログラムにします。プログラムの行50～100で、サブルーチン指定命令の **ON**（オン）のあとの値を、くり返し命令で変えているところがポイントです。

●プログラム ：４種類の図形が次々に出る

```
10 SCREEN 5
20 COLOR 1,1,1
30 FOR S=1 TO 4
40   CLS
50   ON S GOSUB 80,110,170,210
60   FOR I=1 TO 800:NEXT I
70 NEXT S:GOTO 30
80 REM -- ｼｶｸ--
90 LINE (88,67)-(168,147),10,BF
100 RETURN
110 REM --ｻﾝｶｸ--
120 LINE (128,47)-(64,153),10
130 LINE (64,153)-(192,153),10
140 LINE (192,153)-(128,47),10
150 PAINT (128,58),10
160 RETURN
170 REM --ｴﾝ--
180 CIRCLE (128,106),48,10
190 PAINT (144,106),10
200 RETURN
210 REM--ﾀﾞｴﾝ--
220 CIRCLE (128,106),56,10,,,.7
230 PAINT (128,106),10
240 RETURN
```


FOR
S=1 TO 4
Sの値は？
1
2
3
4
グラフィック画面に四角形を描き、色をぬる
グラフィック画面に三角形を描き、色をぬる
グラフィック画面に円を描き、色をぬる
グラフィック画面にだ円を描き、色をぬる
NEXT S



注釈文でプログラムをわかりやすく

前出のプログラムをもう一度見てください。行80などに新しい命令があります。これが注釈命令 **REM**(リマーク) で、プログラム中に注釈文を書くときに使います。

注釈命令は、プログラムの実行にはまったく関係しませんから、どのように使ってもよいのですが、多く使うほど、コンピュータ内の記憶場所も多くとられてしまいます。もちろん、まったく使わなくても、実行結果は同じになります。

ふつう、注釈命令は、サブルーチンの仕事内容や、飛びこし命令 **GOTO**(ゴーツー) の飛び先の目印として使われることが多いようです。これによりプログラムは見やすくなります。

注釈命令は、**REM**(リマーク) のかわりにアポストロフィー(')を使うこともできます。たとえば、「REM--シカク--」と「'--シカク--」は同じ意味として扱われます。

注釈文は、1行だけに書くとは限りません。下の例のように、何行にもわたって注釈文を書くことがあります。ゲームの名前や、ゲームソフトの製作者名、製作年月日を入れるときなどに多く用いられます。

基本形

REM(リマーク)␣注釈文

プログラムの中に注釈文を入れる注釈命令

注釈命令文では、コロン(:)も注釈の一部として解釈されるので、コロンで区切ってほかの命令文を加えることはできません。

なお、プログラムの一部を、消さずにしかも実行しないようにするときに、注釈命令が便利です。命令文の先頭にアポストロフィー(')を加えれば、その行全体が注釈文となってしまうので、ほかの部分は実行されても、その部分は実行されません。アポストロフィーをとれば、そのまま命令文に戻りますから、再び実行すると、こんどはその部分も含めて実行されます。

●プログラム:長い注釈文の例

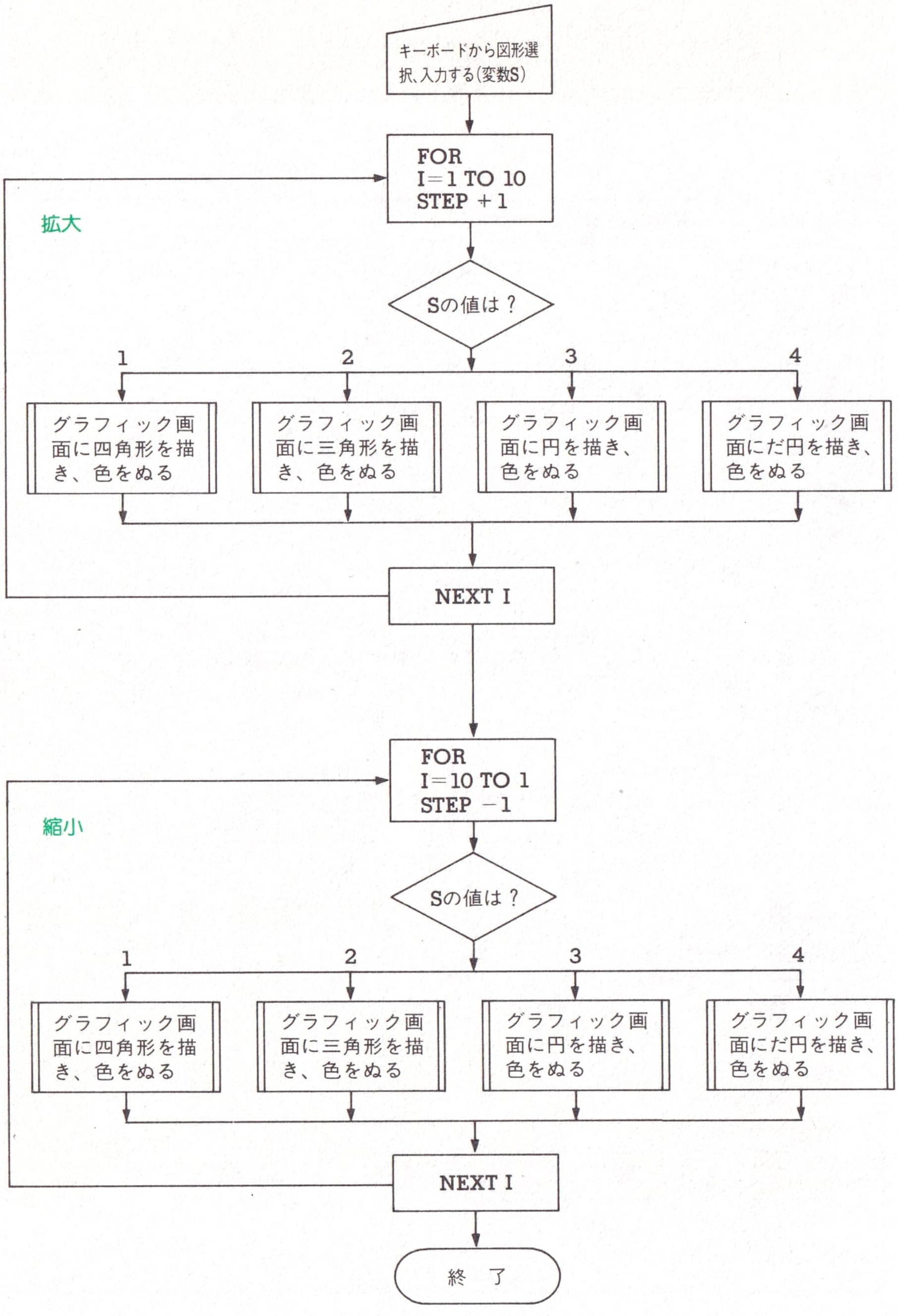
```
10 REM ┌--------------------┐
20 REM |                    |
30 REM | ゲーム:エキサイティング |
40 REM |         カーチェイス |
50 REM | セイサク:ヤマモト タロウ |
60 REM | セイサク年月日:XXXX年 |
70 REM |           6月2日    |
80 REM |                    |
90 REM └--------------------┘
```

図形の拡大、縮小も思いのまま

前出のプログラムに手を加えて、4種類の図形が次々にあらわれ、それが拡大、縮小するようにします。次ページのフローチャートを参考にしてください。

●プログラム：４種類の図形が拡大、縮小する

```
10 INPUT "ｼｶｸ=1,ｻﾝｶｸ=2,ｴﾝ=3,ﾀﾞｴﾝ=4";S
20 SCREEN 5
30 COLOR 1,1,1
40 CLS
50 FOR I=1 TO 10
60   ON S GOSUB 140,190,260,310
70 NEXT I
80 FOR I=10 TO 1 STEP -1
90   ON S GOSUB 140,190,260,310
100 NEXT I
110 COLOR 15,4,7
120 GOTO 10
130 '---ｼｶｸ---
140 LINE (80,55)-(80+I*8,55+8*I),10,B
150 PAINT (81,56),10
160 CLS
170 RETURN
180 '---ｻﾝｶｸ---
190 LINE (128,55)-(128-8*I,55+8*I),10
200 LINE (128-8*I,55+8*I)-(128+8*I,55+8*I),10
210 LINE (128+8*I,55+8*I)-(128,55),10
220 PAINT (128,56),10
230 CLS
240 RETURN
250 '---ｴﾝ---
260 CIRCLE (128,106),I*8,10
270 PAINT (128,106),10
280 CLS
290 RETURN
300 '---ﾀﾞｴﾝ---
310 CIRCLE (128,106),I*8,10,,,.7
320 PAINT (128,106),10
330 CLS
340 RETURN
```


キーボードから図形選択、入力する(変数S)
FOR I=1 TO 10 STEP +1
拡大
Sの値は？
1
2
3
4
グラフィック画面に四角形を描き、色をぬる
グラフィック画面に三角形を描き、色をぬる
グラフィック画面に円を描き、色をぬる
グラフィック画面にだ円を描き、色をぬる
NEXT I
FOR I=10 TO 1 STEP −1
縮小
Sの値は？
1
2
3
4
グラフィック画面に四角形を描き、色をぬる
グラフィック画面に三角形を描き、色をぬる
グラフィック画面に円を描き、色をぬる
グラフィック画面にだ円を描き、色をぬる
NEXT I
終　了

乱数関数を使えば色も大きさもでたらめに

図形を描くプログラムでは、位置の座標、大きさ、色など、すべて変数として値を変えることが可能です。乱数関数 **RND**（ランダム）を使って、これらの変数にでたらめな値を代入できるようにすると、位置も大きさも色も、予想できない形で表示することができます。

基本的な手順は、次のとおりです。

①3 種類の乱数を発生させる。

②それぞれの乱数をX、Y、Cに代入する。

③上記②に基づいて、四角形を描く。

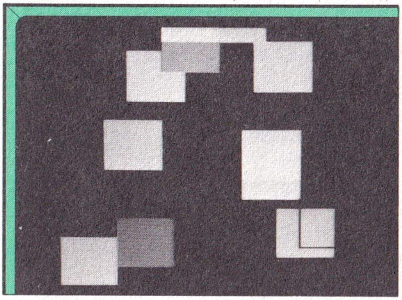
④上記②に基づいて、四角形に色をぬる。

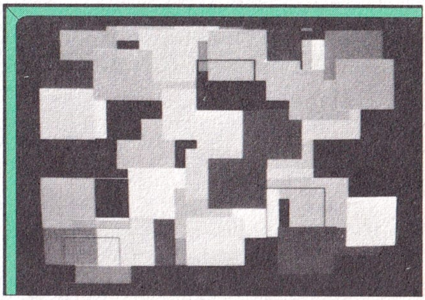
乱数関数で出すでたらめな数の範囲に注意しましょう。画面は 256×212 の座標になります。図形はこれまでどおり横、縦 8 ドットずつ描くことにすると、Xが 0〜31（256÷8）、Yが 0〜26（212÷8）の範囲になります（いちばんはしは使わない）。また、パレット番号は周辺色と同じ色になる 0 は使わないで、1〜15 の範囲が変数Cになります。乱数発生の部分は、プログラムの行 40 です。

●プログラム：さまざまな四角形を描く

```
10 SCREEN 5
20 COLOR 1,1,1
30 CLS
40 X=INT(RND(1)*32):Y=INT(RND(1)*27):C=INT(RND
   (1)*15+1)
50 LINE (X*8,Y*8)-(X*8+40,Y*8+40),C,B
60 PAINT (X*8+1,Y*8+1),C
70 GOTO 40
```

実行





色いろいろ、大きさいろいろの4つの図形

4種類の図形が選べ、選んだ図形がさまざまな位置に、いろいろな色で描かれるプログラムをつくります。前項で説明したことが基本で、それぞれの図形を描く部分をサブルーチンとして考えれば、それほどむずかしくはないでしょう。

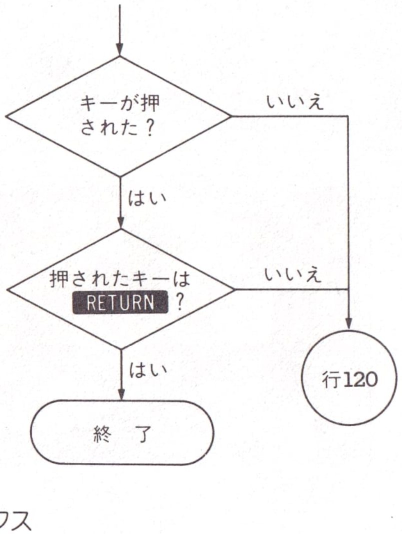
このプログラムは、実行を終わりにするかどうかを指定できる形にしています（行100～110）。行100にある**INKEY$**（インキー・ダラー）は新しい命令で、入力キー関数です。この関数は、キーボードで打ちこんだ文字を読みとって、関数の値とします。この関数が実行されているときに、たとえば[Z]のキーを押すと、Zがこの関数の値になります。ふつうは「A$=INKEY$」という形にして、プログラム中に使います。この場合には、**INKEY$**（インキー・ダラー）の値は「A$」に代入されることになります。

どのキーも押されない場合は、**INKEY$**（インキー・ダラー）の値はカラッポ（ヌルストリング）として扱われます。

キーボードのキーを読みとる点では入力命令**INPUT**（インプット）と似ていますが、**INKEY$**（インキー・ダラー）の場合は**INPUT**（インプット）と違い、押したキーが画面上に表示されません。

行110が、押されたキーの内容を判定しています。**IF～THEN**（イフ・ゼン）が2重に使われています。「LEN（A$）<>0」は、「A$に収納されている文字の長さが0でなければ」という意味です。キーが押されていなければ、A$はカラですから、「LEN（A$）」は0となります。つまり、この部分は、キーが押されたかどうかを判定しているわけです。キーが押されていなければ、次の行に進みます。

左の「THEN」の後ろにある「IF ASC（A$）=13」は、A$のキャラクタコードが13かどうかを判定しています。



基本形

インキー・ダラー
INKEY$

押されているキーの文字を関数の値にする入力キー関数

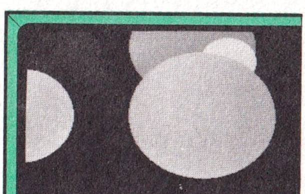
キャラクタコード13は、「入力終了キーが押された状態」をあらわします。入力終了キーが押された場合には、右の「THEN」以下が実行されて、プログラムの実行が終わります。

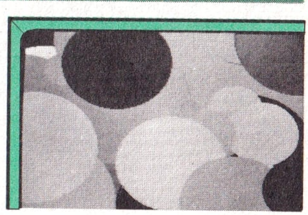
●プログラム：選んだ図形がさまざまに描かれる

```
10 INPUT "ｼｶｸ=1,ｻﾝｶｸ=2,ｴﾝ=3,ﾀﾞｴﾝ=4";S
20 SCREEN 5
30 COLOR 1,1,1
40 CLS
50 X=INT(RND(1)*32):Y=INT(RND(1)*27):C=INT(RND
  (1)*15+1)
60 IF S=1 THEN GOSUB 140:GOTO 100
70 IF S=2 THEN GOSUB 170:GOTO 100
80 IF S=3 THEN GOSUB 220:GOTO 100
90 IF S=4 THEN GOSUB 250:GOTO 100
100 A$=INKEY$
110 IF LEN(A$)<>0 THEN IF ASC(A$)=13 THEN COLOR
   15,4,7:END
120 GOTO 50
130 END
140 LINE (X*8,Y*8)-(X*8+40,Y*8+40),C,B
150 PAINT (X*8+1,Y*8+1),C
160 RETURN
170 LINE (X*8,Y*8)-(X*8-24,Y*8+40),C
180 LINE (X*8-24,Y*8+40)-(X*8+24,Y*8+40),C
190 LINE (X*8+24,Y*8+40)-(X*8,Y*8),C
200 PAINT (X*8,Y*8+1),C
210 RETURN
220 CIRCLE (X*8,Y*8),3*8,C
230 PAINT (X*8,Y*8),C
240 RETURN
250 CIRCLE (X*8,Y*8),3*8,C,,,.7
260 PAINT (X*8,Y*8),C
270 RETURN
```

練習問題13

右のような、さまざまな円を描くプログラムをつくってください(色や大きさは自由)。





LESSON ⑤

パソコン・アニメに挑戦！

絵を動かすにはスプライト機能

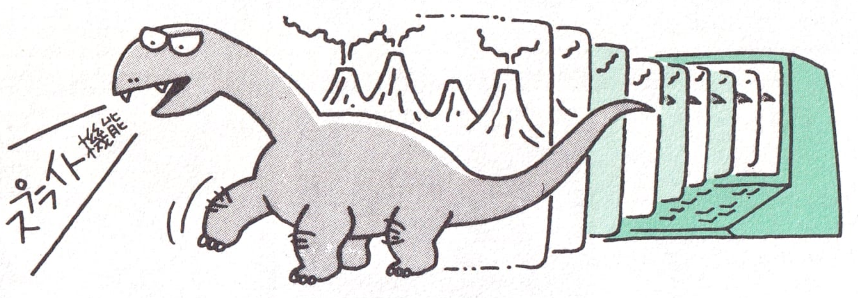
グラフィック画面で図形を動かす方法は、マスターできたと思います。ただ、グラフィック命令で描く直線と円だけでは、自由に絵を描くというわけにはいきません。ビルの前を自動車が走っていく、といった絵を描くのもたいへんです。それにはスプライト機能を使います。

MSX 2 パソコンには、テキスト画面とグラフィック画面のほかに、もうひとつ、スプライト画面というものがあります。これは、画面の前に何枚もの画面が重なっている状態だと考えればよいでしょう。全部で32面あり、それぞれに絵を描くことができます。これが、スプライト機能です。

この機能を使うと、背景と絵を別々の画面に描けるので、背景を動かさずに絵だけを動かすことができます。また、何枚もの絵を重ねると、立体感が出ます。アニメーションやゲームなどには、たいへん便利な機能です。

スプライト機能は、「SCREEN　0」以外のモードで使えます。絵は同時に32個表示できますが、水平方向に同時に描ける個数には下のような制限があります。

モード	同時に描ける絵の個数	制限を超えて指定した場合
SCREEN 1～3	4個	スプライト画面番号の小さい順の4個だけ表示
SCREEN 4～8	8個	スプライト画面番号の小さい順の8個だけ表示



スプライトで絵を描く前に

スプライト機能を使って、スプライト画面に絵を描くには、3つの手順が必要です。

①スプライト使用の条件を設定する（**SCREEN**）。

②スプライト画面に描く絵（スプライト・パターン）を定義する（**SPRITE$**）。

③スプライト・パターンを表示する（**PUT SPRITE**）。

なお、「SCREEN　4～8」では、**COLOR SPRITE** 命令を使って、スプライト・パターンの1ラインごとに色を指定できます。これについては、後述します。

スプライト・パターンは、大きさによって4つに分かれます。まず最初にそれを設定する必要があります。それには、モード設定命令 **SCREEN** を使います。

SCREEN　画面モード，スプライト・パターンの大きさ

画面モードの設定については、すでにお話ししたとおりです(167ページ参照)。スプライト・パターンの大きさは、数値によって次のようになります。

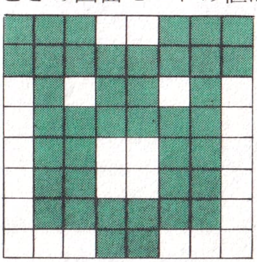
SCREEN ,0………横8×縦8の点で構成する

SCREEN ,1………横8×縦8の点で縦、横を2倍に拡大して表示する

SCREEN ,2………横16×縦16の点で構成する

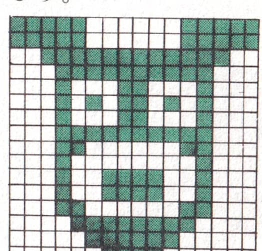
SCREEN ,3………横16×縦16の点で縦、横を2倍に拡大して表示する

画面モードといっしょに設定するのがふつうですから、たとえば「SCREEN 5, 2」とすれば、「SCREEN　5」のビットマップ・モードで、横16×縦16のスプライト・パターンを設定したことになります。先にも説明したように、テキスト画面Ⅰ（SCREEN　0）ではスプライト機能が使えないので、スプライト機能を使うときの画面モードの値は1～8のどれかになります。



横8×縦8





横16×縦16

絵を描いて定義する

次に、手順②のスプライト・パターンの定義に入ります。8×8、または16×16の大きさで考えることにしますから、そのようなマス目（方眼紙）を用意してください。そして、自分が描きたいパターンを考えて、マス目をつぶしていきます。たとえば、8×8なら、下図のようなぐあいです。

これを下のようなプログラムにするには、まず、ぬりつぶしたマス目を1、ぬらないところを0とします。各行（S1～8）ごとに「CHR$(&B」をつけて1行ごとにプログラム化し、それらをプラス記号でつないで動画定義変数 **SPRITE$**（スプライト・ダラー）の右辺に置けば、定義ができたことになります。

基本形のうち、パターン番号は、スプライト・パターンの大きさによって制限があります。8×8のときには0～255（256種類）、16×16のときには0～63（64種類）です。いいかえれば、それぞれの大きさのときに、制限数までパターンの種類を登録しておくことができるというわけです。また、動画定義変数 **SPRITE$**（スプライト・ダラー）は、文字型です。関係する変数にはすべてダラー記号（$）をつけて、文字型に統一しておく必要があります。

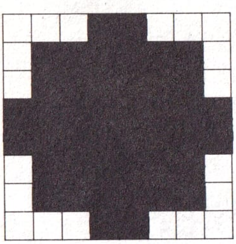
基本形

SPRITE$（スプライト・ダラー）**（パターン番号）**

スプライト・パターンを定義する動画定義変数

プログラムでは、たとえば行30～100が各行の指定で、行110～120でそれを定義し、パターン番号0として登録しています。なお、パターンを定義する行110と行120は、次のようにまとめることもできます。

```
SPRITE$(0)=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
```



➡

0	0	0	1	1	0	0	0	S1
0	1	1	1	1	1	1	0	S2
0	1	1	1	1	1	1	0	S3
1	1	1	1	1	1	1	1	S4
1	1	1	1	1	1	1	1	S5
0	1	1	1	1	1	1	0	S6
0	1	1	1	1	1	1	0	S7
0	0	0	1	1	0	0	0	S8

➡

```
S1$=CHR$(&B 00011000)
S2$=CHR$(&B 01111110)
S3$=CHR$(&B 01111110)
S4$=CHR$(&B 11111111)
S5$=CHR$(&B 11111111)
S6$=CHR$(&B 01111110)
S7$=CHR$(&B 01111110)
S8$=CHR$(&B 00011000)
```

▲パターンの定義（8×8）

●プログラム

```
10 SCREEN 5,1
20 COLOR 1,1,1:CLS
30 S1$=CHR$(&B00011000)
40 S2$=CHR$(&B01111110)
50 S3$=CHR$(&B01111110)
60 S4$=CHR$(&B11111111)
70 S5$=CHR$(&B11111111)
80 S6$=CHR$(&B01111110)
90 S7$=CHR$(&B01111110)
100 S8$=CHR$(&B00011000)
110 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
120 SPRITE$(0)=S0$
```

C/O/L/U/M/N

◉コンピュータ・グラフィックスの世界◉

最近、テレビコマーシャルなど、コンピュータで製作したアニメーションが目につく。コンピュータで描いた絵のことをコンピュータ・グラフィックスということは知っているね。いま、この世界はどんどん広がっている。

テレビのアニメーションばかりでなく、映画もコンピュータでつくられるものが多くなってきた。アニメの画面が簡単にできるのが、増えた理由だ。

アニメーションで人間や動物が自然に動いているように見せるには、1秒間に24枚の絵を描かなくてはならない。その24枚の絵は、少しずつずらして描く必要があるから、人間がやるとすごく手間がかかる。ところが、コンピュータを使うと、1秒間の最初の1枚と最後の1枚の計2枚だけを人間が描けばすむ。残りの22枚は、コンピュータで描くことができるのだ。

アニメーションより大がかりなものでは、レーザー光線を使った絵がある。小さな画面ではなく、スキー場のスロープやコンサートのステージなどに出すのだからスゴイ。東京ディズニーランドのシンデレラ城の壁に、レーザー光線で描かれたミッキーマウスやティンカーベルなどが踊っているのを見た人もいるだろう。あれにもコンピュータとレーザー光線が使われている。これからどんなものが出てくるか、とても楽しみだ。



パターンを表示するPUT SPRITE

それでは、定義したスプライト・パターンを表示してみましょう。それには、動画表示命令 **PUT SPRITE** を使います。たとえば、次のようにスプライト・パターンを表示するとしましょう。

①スプライト画面の5番めの、
②座標（120, 40）に、
③赤（パレット番号8）で、
④パターン番号0で定義したスプライト・パターン。

```
PUT SPRITE 5, (120, 40), 8, 0
           ↑      ↑      ↑  ↑
           ①      ②      ③  ④
```

②の座標は、横が－32～255、縦が－32～191の範囲で指定します。縦の座標を208(「SCREEN　4～8」では216)にすると、指定したスプライト画面よりパターン番号の多いスプライト・パターンがすべて消えます。また、縦の座標を209(「SCREEN　4～8」では217)にすると、指定したスプライト画面に表示されていたスプライト・パターンだけが消えます。

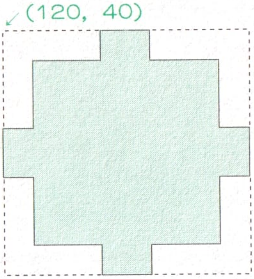
ここで指定している座標は、正確にいうと、スプライト・パターンの左上角の位置になります。上述の例でいえば、たとえば下図のとおりです。この意味からも、最初はスプライト・パターンの大きさに従ったマス目をつくり、そこで絵を決めていくほうがよいでしょう。表示位置の失敗がなくなるはずです。

基本形のうち、数式①はスプライト画面の番号、数式②はパターン番号です。したがって数式①は0～31、数式②は0～255（8×8のとき）または0～63（16×16のとき）の範囲内で指定することになります。

基本形

PUT　SPRITE␣数式①,（座標 X, Y）, パレット番号, 数式②

定義したスプライト・パターンを表示する動画表示命令

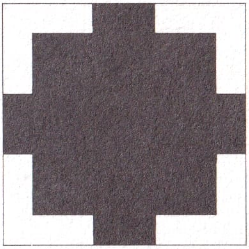


円、三角、四角をスプライト画面に

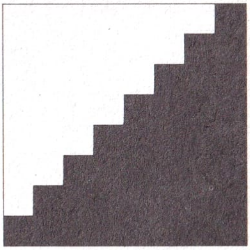
スプライト機能がひととおり頭に入ったところで、これから、「スプライト画面上を円と三角がでたらめに動き、衝突すると四角になる」という課題にとりくみます。

まず、円、三角、四角を定義し、表示してみることにします。パターンは次のような形にします。円は正確には描けないので、似たような形でがまんすることにしましょう。

円



三角



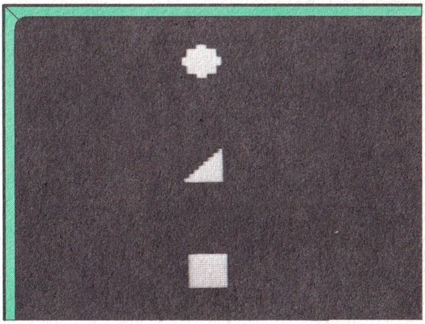
四角



画面モードは「SCREEN　5」、スプライト・パターンの大きさは8×8の拡大にします。そして、それぞれの座標と色は次のように決めます。

・円……(122, 40) に暗い青で描く
・三角……(122, 90) に赤で描く
・四角……(122, 140) に緑で描く

実行結果は、写真のようになります。

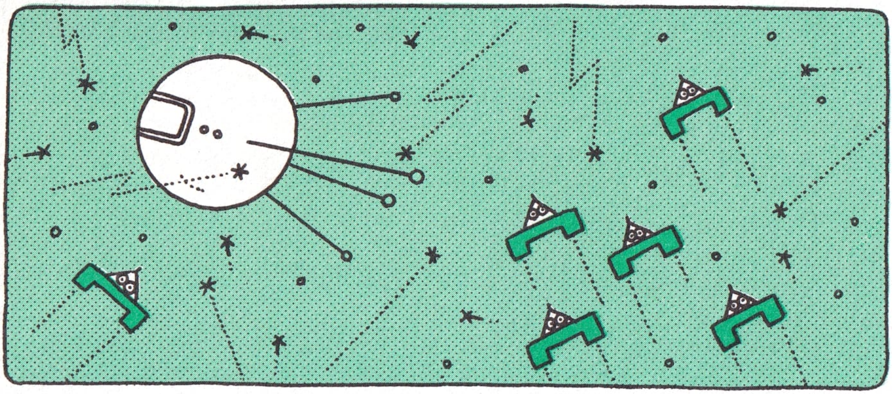


実行

●プログラム：円、三角、四角をスプライト画面に表示する

```
10 SCREEN 5,1
20 COLOR 1,1,1:CLS
30 S1$=CHR$(&B00011000)
40 S2$=CHR$(&B01111110)
50 S3$=CHR$(&B01111110)
60 S4$=CHR$(&B11111111)
70 S5$=CHR$(&B11111111)
```

```
80 S6$=CHR$(&B01111110)
90 S7$=CHR$(&B01111110)
100 S8$=CHR$(&B00011000)
110 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
120 SPRITE$(0)=S0$
130 S1$=CHR$(&B00000001)
140 S2$=CHR$(&B00000011)
150 S3$=CHR$(&B00000111)
160 S4$=CHR$(&B00001111)
170 S5$=CHR$(&B00011111)
180 S6$=CHR$(&B00111111)
190 S7$=CHR$(&B01111111)
200 S8$=CHR$(&B11111111)
210 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
220 SPRITE$(1)=S0$
230 S1$=CHR$(&B11111111)
240 S2$=CHR$(&B11111111)
250 S3$=CHR$(&B11111111)
260 S4$=CHR$(&B11111111)
270 S5$=CHR$(&B11111111)
280 S6$=CHR$(&B11111111)
290 S7$=CHR$(&B11111111)
300 S8$=CHR$(&B11111111)
310 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
320 SPRITE$(2)=S0$
330 PUT SPRITE 0,(122,40),4,0
340 PUT SPRITE 1,(122,90),8,1
350 PUT SPRITE 2,(122,140),2,2
360 GOTO 360
```



スプライト・パターンを動かそう

スプライト・パターンを動かしてみましょう。少しずつアニメーションが実現してきます。グラフィック画面で図形を動かすには、座標を変化させました。スプライト画面でも、その原理は同じです。ただし、動きはずっとスムーズに見えます。その秘密は、「1枚のスプライト画面には1つのパターンしか表示されない」という原理にあります。パターンの表示位置を少しずつずらして表示するようにすると、新しい位置で表示したとたんに前に表示されていたパターンが消えるので、あたかも動いているように見えるのです。いうまでもなく、前の表示を消す処理は不要です。

座標を動かすのですから、問題になるのは **PUT**(プット) **SPRITE**(スプライト) 文の座標を指定するところです。ここを変数にして、適当に値を変えれば、パターンが動いているように見えます。

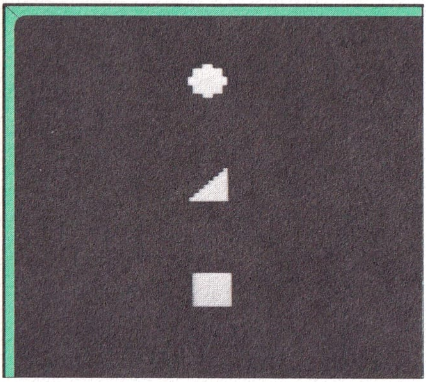
下のプログラムでは、横に動かしています。行330～370の **FOR～NEXT**(フォー・ネクスト) により、各パターンのIの値を0→255に変えることで、パターンは左から右へ移動していきます。また、行380～420の **FOR～NEXT**(フォー・ネクスト) では、Iの値が255→0と変化するので、逆に右から左へ動きます。

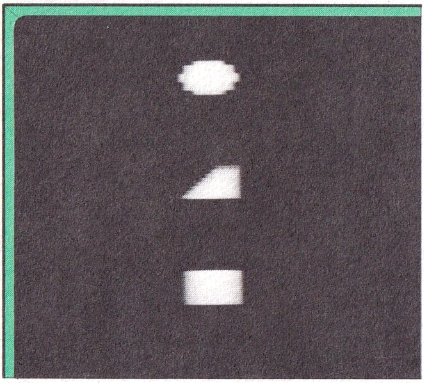
●プログラム：スプライト・パターンを横に動かす

```
10 SCREEN 5,1
20 COLOR 1,1,1:CLS
30 S1$=CHR$(&B00011000)
40 S2$=CHR$(&B01111110)
50 S3$=CHR$(&B01111110)
60 S4$=CHR$(&B11111111)
70 S5$=CHR$(&B11111111)
80 S6$=CHR$(&B01111110)
90 S7$=CHR$(&B01111110)
100 S8$=CHR$(&B00011000)
110 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
120 SPRITE$(0)=S0$
130 S1$=CHR$(&B00000001)
140 S2$=CHR$(&B00000011)
150 S3$=CHR$(&B00000111)
160 S4$=CHR$(&B00001111)
170 S5$=CHR$(&B00011111)
```

```
180 S6$=CHR$(&B00111111)
190 S7$=CHR$(&B01111111)
200 S8$=CHR$(&B11111111)
210 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
220 SPRITE$(1)=S0$
230 S1$=CHR$(&B11111111)
240 S2$=CHR$(&B11111111)
250 S3$=CHR$(&B11111111)
260 S4$=CHR$(&B11111111)
270 S5$=CHR$(&B11111111)
280 S6$=CHR$(&B11111111)
290 S7$=CHR$(&B11111111)
300 S8$=CHR$(&B11111111)
310 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
320 SPRITE$(2)=S0$
330 FOR I=0 TO 255
340   PUT SPRITE 0,(I,49),4,0
350   PUT SPRITE 1,(I,99),8,1
360   PUT SPRITE 2,(I,149),2,2
370 NEXT I
380 FOR I=255 TO 0 STEP -1
390   PUT SPRITE 0,(I,49),4,0
400   PUT SPRITE 1,(I,99),8,1
410   PUT SPRITE 2,(I,149),2,2
420 NEXT I
430 GOTO 330
440 END
```

実行





円と三角が衝突して四角に

さあ、いよいよ本題にとりかかることにしましょう。円と三角が衝突するというのは、人の目にはそう見えますが、実際には、円と三角は別々のスプライト画面に描かれているのですから、衝突とはいえません。重なっただけです。「円と三角が重なったときに四角になる」という課題のポイントは、次のとおりです。

①円と三角が重なったことを判断し、
②円と三角を消し、
③四角形を表示する。

①では、新しい命令が必要です。**ON　SPRITE　GOSUB**という動画割りこみ設定命令です。この命令は、スプライト・パターンが重なったときに、指定した行番号へ飛ぶ命令です。この命令は、プログラムの処理順序にかかわりなく、スプライト・パターンが重なるという現象が起きたときだけ実行されます。このように、特別なことが起きたとき、現在実行している処理を一時停止して、特別な処理を優先して実行させることを「割りこみ」といいます。なお、基本形の「行番号」は、割りこみ後に実行する処理の、最初の行番号です。

また、**ON　SPRITE　GOSUB**と対のように使われるのが、動画割りこみ命令**SPRITE ON**と、動画割りこみ禁止命令**SPRITE　OFF**です。**SRPITE　ON**は、スプライトが衝突したことによる割りこみを許可する命令で、これを実行しておくと、**ON SPRITE　GOSUB**文で設定した特別な処理が実行されます。**SPRITE　OFF**は、スプライト衝突による割りこみを禁止するので、この命令の実行後は、スプライト・パターンが重なっても割りこみは起こりません。

基本形

ON　SPRITE　GOSUB␣行番号
スプライト・パターンの衝突による割りこみを設定する動画割りこみ設定命令

SPRITE　ON
スプライト・パターンの衝突による割りこみを許可する動画割りこみ命令

SPRITE　OFF
スプライト・パターンの衝突による割りこみを禁止する動画割りこみ禁止命令

②は、動画表示命令 **PUT** **SPRITE** 文中の座標の位置を、指定範囲外に指定することで、表示されないようにします（行640～650）。

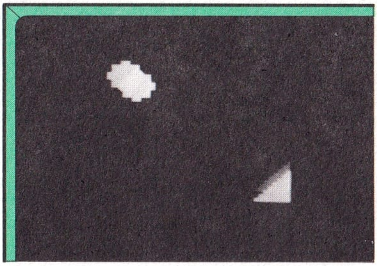
③は、簡単です。行660を見てください。

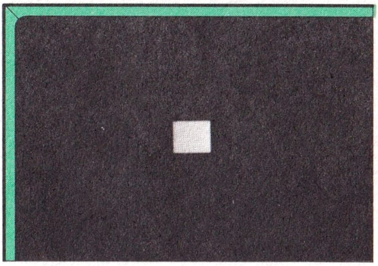
●プログラム：円と三角が衝突して四角になる

```
10 SCREEN 5,1
20 COLOR 1,1,1:CLS
30 S1$=CHR$(&B00011000)
40 S2$=CHR$(&B01111110)
50 S3$=CHR$(&B01111110)
60 S4$=CHR$(&B11111111)
70 S5$=CHR$(&B11111111)
80 S6$=CHR$(&B01111110)
90 S7$=CHR$(&B01111110)
100 S8$=CHR$(&B00011000)
110 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
120 SPRITE$(0)=S0$
130 S1$=CHR$(&B00000001)
140 S2$=CHR$(&B00000011)
150 S3$=CHR$(&B00000111)
i60 S4$=CHR$(&B00001111)
170 S5$=CHR$(&B00011111)
180 S6$=CHR$(&B00111111)
190 S7$=CHR$(&B01111111)
200 S8$=CHR$(&B11111111)
210 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
220 SPRITE$(1)=S0$
230 S1$=CHR$(&B11111111)
240 S2$=CHR$(&B11111111)
250 S3$=CHR$(&B11111111)
260 S4$=CHR$(&B11111111)
270 S5$=CHR$(&B11111111)
280 S6$=CHR$(&B11111111)
290 S7$=CHR$(&B11111111)
300 S8$=CHR$(&B11111111)
310 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
320 SPRITE$(2)=S0$
330 DIM VX(4),VY(4)
340 VX(1)=0:VX(2)=1:VX(3)=0:VX(4)=-1
350 VY(1)=-1:VY(2)=0:VY(3)=1:VY(4)=0
360 ON SPRITE GOSUB 630
370 SPRITE ON
```

```
380 X1=110:X2=140
390 Y1=70:Y2=100
400 'マル ヲ ウゴカス
410 TY=INT(RND(1)*4+1)
420 TX=INT(RND(1)*4+1)
430 TX=X1+VX(TX)*16
440 TY=Y1+VY(TY)*16
450 IF TX<40 THEN 380
460 IF TY<40 THEN 380
470 IF TX>200 THEN 380
480 IF TY>140 THEN 380
490 X1=TX:Y1=TY
500 PUT SPRITE 0,(X1,Y1),4,0
510 'サンカク ヲ ウゴカス
520 TY=INT(RND(1)*4+1)
530 TX=INT(RND(1)*4+1)
540 TX=X2+VX(TX)*16
550 TY=Y2+VY(TY)*16
560 IF TX<40 THEN<40 THEN 380
570 IF TY<40 THEN 380
580 IF TX>200 THEN 380
590 IF TY>140 THEN 380
600 X2=TX:Y2=TY
610 PUT SPRITE 1,(X2,Y2),8,1
620 GOTO 410
630 SPRITE OFF
640 PUT SPRITE 0,(X1,217),,0
650 PUT SPRITE 1,(X2,217),,1
660 PUT SPRITE 2,(X1,Y1),2,2
670 FOR I=1 TO 1000:NEXT I
680 PUT SPRITE 2,(X1,217),,2
690 SPRITE ON
700 RETURN 380
```

実行





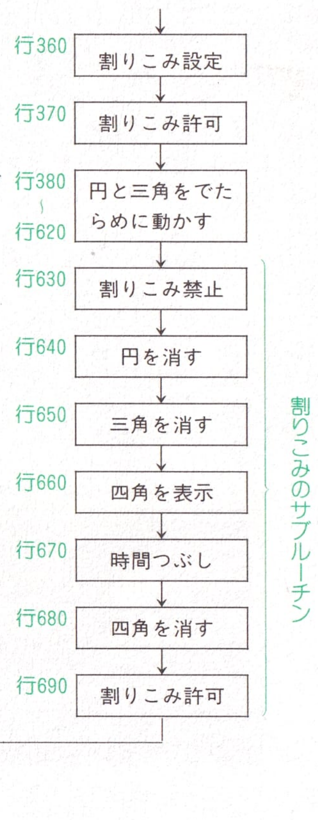
割りこみ処理の注意点

前ページの「円と三角が衝突して四角になる」プログラムをもう一度見てみましょう。ここで注目してほしいのは、円と三角が衝突したときの割りこみ処理です。この処理に直接関係するのは、行360、行370、行630、行690です。とくに**SPRITE ON**と**SPRITE OFF**の位置に注意してください。

動画割りこみ設定命令**ON SPRITE GOSUB**はスプライト・パターンが衝突したときの飛び先を設定するだけです。ですから、この命令が実行され、衝突が起こったとしても、指定の行に飛ぶわけではありません。**SPRITE ON**が実行されてはじめて、指定の行に飛びます。

行360のすぐ次の行370に**SPRITE ON**を書いてあるのはこのためです。ふつうは、このように、**ON SPRITE GOSUB**と**SPRITE ON**はいつもワンセットとして考えるとよいでしょう。行360と行370をマルチステートメントにして、

「ON SPRITE GOSUB 630:SPRITE ON」と書くことも多いようです。



次に、スプライト・パターンの衝突が起こったときの飛び先に注目。行630でSPRITE OFFになっています。これは、行630に飛んだ時点で、すでに始まっている衝突による処理の実行中に、別の割りこみが起こらないための処置です。

衝突した場合の処理は行640～680でいちおう終わりで、あとは再びもとのプログラム（行380）に戻って、円と三角を表示して動かしますが、次の、円と三角の衝突による割りこみに備えて、行690の**SPRITE ON**が必要です。行380に戻ってからは、行370は実行されることはないので、行690がないと、衝突があっても割りこみが起こりません。

COLOR SPRITEで細かに彩色

MSXベーシックのバージョン2.0では、スプライト機能に関して、便利な命令が増えました。それが、これから紹介する動画彩色命令**COLOR SPRITE**です。これまでは、動画表示命令**PUT SPRITE**でパターンの色を指定しました。この場合は、パターン全体に1色をぬるだけでしたが、この命令を使うと、スプライト・パターンの横1列ごとに色を指定することができます。

たとえば、次のようにすると、パレットが初期状態であれば、スプライト画面3番めのパターンの1列めが明るい緑、2列めが明るい赤になります。

```
COLOR SPRITE$(3)=CHR$(3)+CHR$(9)
```

この命令は、画面モードが「SCREEN　4～8」のときだけ使えます。

基本形の「式」は「CHR$（パレット番号）」をプラス記号（＋）でつないだものです。そして、並んだ**CHR$**の順番が、パターンの列の順番に対応しています。つまり、1番めの**CHR$**は第1列め、2番めの**CHR$**は第2列め、3番めの**CHR$**は第3列め……の色を指定することになります。ですから、もし、8×8のスプライト・パターンで、全列について別々の色を指定するなら、「＝」の右側には**CHR$**が8個並ぶことになります。指定してない列の色は、以前に実行したパターンの色がそのまま残ります。

「SCREEN　8」の場合は、パレット番号ではなく、固定のカラーコードを使います（185ページ参照）。

なお、**COLOR SPRITE**のあと、動画表示命令**PUT SPRITE**で色を指定すると、前の指定は無効になってしまうので注意が必要です。これをさけるために、**PUT SPRITE**での色指定は省略します。

基本形

COLOR SPRITE（スプラト画面番号）＝式

CHR$で与えられたキャラクタコードによって、スプライト・パターンの各ラインの色を設定する動画彩色命令。文字として扱われる右辺に合わせるため、通常は$をつけて「COLOR　SPRITE$」とする

カラフルなパターンをつくる

では、動画彩色命令 **COLOR SPRITE** を使ってみましょう。衝突したあとのパターンをカラフルで動きのある感じにしました。行 730 で **COLOR SPRITE** を使っています。各ラインの色は、行 700 ～ 720 の **FOR ～NEXT** で変えています。

●プログラム：カラフルなパターンを描く

```
10 T=RND(-TIME)
20 COLOR 1,1,1
30 SCREEN 5,1
40 S1$=CHR$(&B00011000)
50 S2$=CHR$(&B01111110)
60 S3$=CHR$(&B01111110)
70 S4$=CHR$(&B11111111)
80 S5$=CHR$(&B11111111)
90 S6$=CHR$(&B01111110)
100 S7$=CHR$(&B01111110)
110 S8$=CHR$(&B00011000)
120 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
130 SPRITE$(0)=S0$
140 S1$=CHR$(&B00000001)
150 S2$=CHR$(&B00000011)
160 S3$=CHR$(&B00000111)
170 S4$=CHR$(&B00001111)
180 S5$=CHR$(&B00011111)
190 S6$=CHR$(&B00111111)
200 S7$=CHR$(&B01111111)
210 S8$=CHR$(&B11111111)
220 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
230 SPRITE$(1)=S0$
240 S1$=CHR$(&B11111111)
250 S2$=CHR$(&B11111111)
260 S3$=CHR$(&B11111111)
270 S4$=CHR$(&B11111111)
280 S5$=CHR$(&B11111111)
290 S6$=CHR$(&B11111111)
300 S7$=CHR$(&B11111111)
310 S8$=CHR$(&B11111111)
320 S0$=S1$+S2$+S3$+S4$+S5$+S6$+S7$+S8$
330 SPRITE$(2)=S0$
340 DIM VX(4),VY(4)
```

```
350 VX(1)=0:VX(2)=1:VX(3)=0:VX(4)=-1
360 VY(1)=-1:VY(2)=0:VY(3)=1:VY(4)=0
370 ON SPRITE GOSUB 640
380 SPRITE ON
390 X1=110:X2=140
400 Y1=70:Y2=100
410 'マル ヲ ウゴカス
420 TY=INT(RND(1)*4+1)
430 TX=INT(RND(1)*4+1)
440 TX=X1+VX(TX)*16
450 TY=Y1+VY(TY)*16
460 IF TX<40 THEN 390
470 IF TY<40 THEN 390
480 IF TX>200 THEN 390
490 IF TY>140 THEN 390
500 X1=TX:Y1=TY
510 PUT SPRITE 0,(X1,Y1),4,0
520 ' サンカク ヲ ウゴカス
530 TY=INT(RND(1)*4+1)
540 TX=INT(RND(1)*4+1)
550 TX=X2+VX(TX)*16
560 TY=Y2+VY(TY)*16
570 IF TX<40 THEN 390
580 IF TY<40 THEN 390
590 IF TX>200 THEN 390
600 IF TY>140 THEN 390
610 X2=TX:Y2=TY
620 PUT SPRITE 1,(X2,Y2),8,1
630 GOTO 420
640 SPRITE OFF
650 PUT SPRITE 0,(X1,217),,0
660 PUT SPRITE 1,(X2,217),,1
670 PUT SPRITE 1,(X2,217),,1
680 PUT SPRITE 2,(X1,Y1),2,2
690 FOR J=1 TO 8
700  FOR K=J TO J+15
710   S1$=S1$+CHR$(K)
720  NEXT K
730  COLOR SPRITE$(2)=S1$
740  S1$=""
750 NEXT J
760 FOR I=1 TO 500:NEXT I
770 PUT SPRITE 2,(X1,217),,2
780 SPRITE ON
790 RETURN 390
```

練習問題の解答と解説

（注） 同じ文字を表示するにも、機種によってキー操作が違います。以下の解答では本書で用いたキーを使い、「イ」を出す、などと説明しています。キーボード、マニュアルを参照して、自分の機種に合った、正しいキー操作をしてください。

●練習問題 1 ―――― 35ページ

カタカナを出すときは、かな キーと CAPS LOCK キーをロックしておいて、その文字のキーを押します。

① 1-→でカーソルを左側の「オ」まで移動　2-「イ」を出す

② 1-→でカーソルを左側の「エ」まで移動　2-DEL キーを押す

③ 1-→でカーソルを「カ」まで移動　2-DEL キーを押す

●練習問題 2 ―――― 37ページ

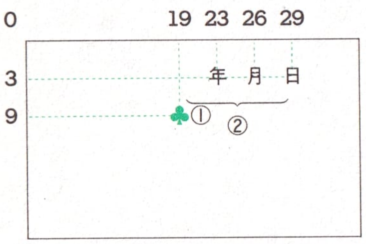
カタカナを出すときは、かな キーと CAPS LOCK キーをロックしておいて、その文字のキーを押します。

① 1-→でカーソルを「ソ」に移動　2-INS キーを1回押す　3-SHIFT キーを押しながら「ー」（音をのばす記号）を出す　4-→でカーソルを「コ」に移動　5-INS キーを押す　6-「ナ」と「ル」を出す

② 1-→でカーソルを「ン」の左側に移動　2-「ヒ」「゜」（半濁点）を出したあと、SHIFT キーを押しながら「ュ」を出す　3-SHIFT キーを押しながら「ー」を出す　4-「タ」を出す

●練習問題 3 ―――― 67ページ

一例として、図のように入力します。



① 1-「LOCATE　19, 9」とキーインし、RETURN キーを押す　2-GRAPH キーを押しながら「♣」を出す

② 1-「LOCATE　23, 3」とキーインし、RETURN キーを押す　2-GRAPH キーを押しながら「年」を出す　3-「LOCATE　26, 3」とキーインし、RETURN キーを押す　4-GRAPH キーを押しながら「月」を出す　5-「LOCATE　29, 3」とキーインし、RETURN キーを押す　6-GRAPH キーを押しながら「日」を出す

〈ポイント〉 GRAPH キーにはロック機能がないので、そのつど押しながら表示しようとする文字のキーを押すこと。

●練習問題 4 ―― 79ページ

①

```
10 C$="●"
20 X=0:Y=17
30 FOR I=0 TO 17
40      LOCATE X,Y:PRINT C$
50      LOCATE X,Y:PRINT " "
60      X=X+1:Y=Y-1
70 NEXT I
80 END
```

②

```
10 C$="●"
20 X=28:Y=16
30 FOR I=0 TO 13
40      LOCATE X,Y:PRINT C$
50      LOCATE X,Y:PRINT " "
60      X=X-1:Y=Y-1
70 NEXT I
80 END
```

③

```
10 C$="●"
20 X=28:Y=0
30 FOR I=0 TO 17
40      LOCATE X,Y:PRINT C$
50      LOCATE X,Y:PR    " "
60      X=X-1:Y=Y+1
70 NEXT I
80 END
```

〈ポイント〉 くり返し命令の、終値のとり方に注意。座標の範囲を超えてしまったら、予想どおりいかない。

●練習問題 5 ―― 85ページ

①

```
10 C$="●"
20 X=10
30 Y=INT(RND(1)*18)
40 LOCATE X,Y:PRINT C$
50 FOR I=1 TO 200
60 NEXT I
70 GOTO 20
```

②

```
10 C$="●"
20 X=INT(RND(1)*29)
30 Y=15
40 LOCATE X,Y:PRINT C$
50 FOR I=1 TO 200
60 NEXT I
70 GOTO 20
```

〈ポイント〉 記号を消しながら動かすには、「65 LOCATE X, Y : PRINT〝 〟」を加える。行50～60は時間つぶし。

●練習問題 6 ―― 110ページ

①

```
10 A=56
20 B=27
30 C=2
40 D=A+B*C
50 PRINT A;"+";B;"*";C;"=";D
60 END
```

②

```
10 A=64
20 B=2
30 C=32
40 D=51
50 E=A/B+C-D
60 PRINT A;"/";B;"+";C;"-";D;"=";E
70 END
```

〈ポイント〉 演算式をそのまま表示命令のあとに持ってくることもできる。①でいえば、「PRINT A ;"+" ;B ;"*" ;C ;"=" ;A+B*C」となる。こうすると行40が不要になり、プログラムは短くなるが、かえってわかりにくいということになる。

●練習問題 7

115ページ

①

```
10 INPUT "ダスウハ";B
20 INPUT "アンダスウハ";H
30 R=H/B*10
40 PRINT "ダリツハ";R
50 END
```

②

```
10 INPUT "ガンキンハ";G
20 INPUT "リリツハ";R
30 FOR I=1 TO 10
40     PRINT G*(1+R/100)^I
50 NEXT I
60 END
```

〈ポイント〉 ①について、ふつう、打率はせいぜい小数点以下4位まででであらわされることが多い(毛(もう)まで)。このままだとそれ以下も計算される。そこで、毛(もう)の単位までしか必要ないのなら、「H/B」に10^4を乗じたうえで、整数関数INT(インテジャー)を使い「R=INT(H/B*10^4)」とすれば、小数点5位以下が切りすてられる。

また、毛(もう)以下を四捨五入するなら「R=INT(H/B*10^4+0.5)」でよい。

●練習問題 8

133ページ

A

```
10 A=INT(RND(1)*99+1)
20 B=INT(RND(1)*99+1)
30 INPUT "モンダイセンタク タスー1,ヒクー2,カケルー3,ワルー4";S
40 ON S GOTO 150,160,170,180
50 READ C$
60 IF S=1 THEN K=A+B:GOTO 100
70 IF S=2 THEN K=A-B:GOTO 100
80 IF S=3 THEN K=A*B:GOTO 100
90 IF S=4 THEN K=A/B:GOTO 100
100 PRINT A;C$;B
110 INPUT C
120 IF K=C THEN PRINT "セイカイデス" ELSE PRINT "マチガイデス コタエハ";K:BEEP
130 INPUT "ツヅケマスカ ハイーy イイエーn";T$
140 IF T$="y" THEN GOTO 10 ELSE END
150 RESTORE 150:DATA +:GOTO 50
160 RESTORE 160:DATA -:GOTO 50
170 RESTORE 170:DATA *:GOTO 50
180 RESTORE 180:DATA /:GOTO 50
```

B

```
10 A=INT(RND(1)*99+1)
20 B=INT(RND(1)*99+1)
30 INPUT "モンダイセンタク タスー1,ヒクー2,カケルー3,ワルー4";S
```

```
40 IF S=1 THEN K=A+B:PRINT A;"+";B;"=":GOTO 80
50 IF S=2 THEN K=A-B:PRINT A;"-";B;"=":GOTO 80
60 IF S=3 THEN K=A*B:PRINT A;"*";B;"=":GOTO 80
70 IF S=4 THEN K=A/B:PRINT A;"/";B;"=":GOTO 80
80 INPUT C
90 IF K=C THEN PRINT "ｾｲｶｲﾃﾞｽ" ELSE PRINT "ﾏﾁｶﾞｲﾃﾞｽ ｺﾀｴﾊ";K:BEEP
100 INPUT "ﾂﾂﾞｹﾏｽｶ ﾊｲ-y ｲｲｴ-n";T$
110 IF T$="y" THEN GOTO 10 ELSE END
```

〈ポイント〉 Aの解答例は、読みこみ命令 **READ〜DATE**（リード・データ）を使ったプログラム、Bは使わないプログラムで、どちらも正解。なお、除算の場合、答えが小数点以下まで続く場合があり、そこまで暗算で答えを出すのは無理なので、整数関係 **INT**（インテジャー）を使い、小数点以下を切りすてるようにしてある(K=INT(A/B))。四捨五入するのなら、「K=INT(A/B+0.5)」となる。

●練習問題 9 ―― 142ページ

```
10 DIM A(10)
20 FOR I=1 TO 10
30     INPUT A(I)
40     S=S+A(I)
50 NEXT I
60 FOR I=1 TO 9
70     PRINT A(I);"+";
80 NEXT I
90 PRINT A(10)
100 PRINT "ｺﾞｳｹｲﾊ";S
110 END
```

〈ポイント〉 演算式の表示法に注意。プログラムの行60〜90は、9番めの値までにプラス(＋)をつけ、最後の値は数字だけ表記するようにしてある。

●練習問題 10 ―― 147ページ

```
10 INPUT "ﾃﾞｰﾀﾊｲｸﾂ";N
20 DIM D$(N)
30 FOR I=1 TO N
40     LOCATE 10,1
50     INPUT "ｼﾒｲ";D$(I)
60     LOCATE 10,1
70     PRINT SPC(20)
80 NEXT I
90 END
```

〈ポイント〉 配列変数の添え字を変数にして、入力命令 **INPUT**（インプット）で入力できるようにすればよい。

●練習問題 11 ―― 153ページ

```
10 DIM A(15)
20 FOR I=1 TO 15
30     INPUT A(I)
40 NEXT I
50 CLS
60 FOR I=1 TO 14
70     JS=I+1
```

```
80      FOR J=JS TO 15
90          IF A(I)<A(J) THEN W=A(I):A(I)=A(J):A(J)=W
100     NEXT J
110 NEXT I
120 FOR I=1 TO 15
130     PRINT A(I)
140 NEXT I
150 END
```

〈ポイント〉　昇順に並べかえるプログラムを1か所だけ修正すればよい。すなわち、行90の関係演算子「>」を、「<」になおせばよい。

●練習問題 12 ——— 177ページ

①

```
10 SCREEN 5
20 LINE (75,50)-(175,150),1,B
30 CIRCLE(125,100),50,1
40 GOTO 40
```

②

```
10 SCREEN 5
20 LINE (75,50)-(175,100),1,B
22 LINE (125,50)-(75,100),1
24 LINE (125,50)-(175,100),1
30 CIRCLE(125,75),66,1,,,.6
40 GOTO 40
```

〈ポイント〉　この問題の要点は、四角形の対角線の交点と、円またはだ円の中心点とが一致すること。グラフ用紙などに描こうとする図形を書いてから、プログラムをつくるようにしたい。

　また、たとえば①の場合、先に四角形をつくってからそれに内接する円を描いても、あるいはその逆でもよい。

●練習問題 13 ——— 193ページ

```
10 COLOR 1,1,1
20 SCREEN 5
30 X=INT(RND(1)*32)
40 Y=INT(RND(1)*24)
50 R=INT(RND(1)*10)
60 C=INT(RND(1)*16)
70 CIRCLE (X*8,Y*8),R*8,C
80 PAINT  (X*8,Y*8),C
90 GOTO 30
```

〈ポイント〉　中心の座標や半径は自由。限定された範囲にだけ表示したり、一定の大きさの円だけを描くようにしたり、自分で課題をいろいろ設定してみるとよい。

エラーメッセージ一覧

比較的よく出てくるものだけを選んであります。

メッセージ	意　味	例・ポイント
ディビジョン バイ ゼロ Division by zero	①0による割り算を行なおうとした ②0に対して負のべき乗を行なった(例：0^{-5})	10　C=0 20　B=A/C 30　PRINT　B ◆割り算の変数の値に要注意
イリーガル ファンクション コール Illegal function call	①命令や関数の使い方が違う ②指定する数値などが使用可能な範囲を超えている	10　FOR　I=1　TO　400 20　X$=CH$(I) 30　NEXT　I ◆I の値が大きすぎる
ミッシング オペランド Missing operand	必要な指定がない	LOCATE　10, ◆コンマのあとの指定もれ
ネクスト ウィズアウト フォー NEXT without FOR	NEXT 文に対応する FOR 文がない	10　PRINT　"*" 20　PRINT　"　" 30　NEXT　I ◆入れ子のループのときに注意
アウト オブ データ Out of DATA	READ 文で読みこませるはずのデータがない	10　READ　A,　B,　C 20　PRINT　A,　B,　C 30　DATA　5,　7 ◆変数とデータの数，型の不一致に注意
アウト オブ メモリ Out of memory	プログラムが長すぎたり、配列が大きすぎたりして、メモリが足りなくなった	◆プログラムが長すぎないか ◆配列（DIM 文）が大きすぎないか ◆GOSUB、FOR～NEXT が多すぎないか
アウト オブ ストリング スペース Out of string space	文字列変数用のメモリが足りなくなった	◆ 1つの文字変数に、256 文字以上も入力していないか
オーバフロウ Overflow	演算結果や入力された数値が、型に応じた許容範囲を超えた	◆たとえば整数の演算で、答えが±32768を超えた場合

メッセージ	意　味	例・ポイント
リディメンションド　アレイ Redimensioned array	①配列変数を二重に定義しようとした ②配列変数の添え字の最大値が 10 になったあとで、配列を宣言しようとした	10 DIM A(50) 20 DIM A(70) ◆2回定義している
リターン　ウィズアウト　ゴーサブ RETURN without GOSUB	GOSUB 文と RETURN 文が、正しく対応していない	80 GOTO 110 ⋮ 110 (サブルーチン) 120 RETURN ◆サブルーチンへ GOTO で飛んでいる
サブスクリプト　アウト　オブ　レインジ Subscript out of range	①指定した配列変数の添え字の値が、DIM 文で宣言した値より大きい ②DIM 文で定義しないで使用した配列が 10 を超えた	10 DIM A(25) ⋮ 50 FOR I=1 TO 70 60 PRINT A(I) 70 NEXT I ◆I の値に注意する
シンタックス　エラー Syntax error	文の書き方が違っている	◆つづりはまちがっていないか ◆区切り記号はあるか ◆変数名は英字で始まっているか ◆代入文の書き方にミスはないか
タイプ　ミスマッチ Type mismatch	データの形式が一致しない	A="KH" A$=LEN(B$) B=CHR$(50) C$=A ◆すべてまちがい
アンディファインド　ライン　ナンバ Undefined line number	指定された行番号が、プログラム中にない	10 PRINT "A" 20 GOTO 15 ◆GOTO 文、GOSUB 文、IF 文などでとくに注意

ふろく 1　パソコン用語の解説

【ア行】

アイ・オー（I/O）　Input（インプット）/Output（アウトプット）の略で入出力のこと。インプットは入力、アウトプットは出力のことで、コンピュータの重要な機能。一般にキーボード、カセットテープなどから入力し、ディスプレイ、プリンタなどへ出力する。

アイ・シー（IC）　Integrated（インテグレイティッド） Circuits（サーキット）の略で、集積回路のこと。IC には、トランジスタの集積度によって、次のような種類がある。

①SSI（小規模集積回路）……トランジスタ 100 個以下

②MSI（中規模集積回路）……トランジスタ 100～1000 個

③LSI（大規模集積回路）……トランジスタ 1000～10 万個

④VLSI（超 LSI）……トランジスタ 10 万個以上

アナログ　長さ、角度、あるいは電流や電圧のように、連続する量を相似的に表現、または測定すること。文字盤、長針、短針のある時計はアナログ時計といわれる。デジタルと対比される。

アルゴリズム　何かの問題を解決するための手順のこと。パソコンにおいては、パソコンにやらせたいことを箇条書きにした手順のこと。

インターフェース　装置と装置をつなぐときに、2 つの装置の違い（電圧とか作動速度など）を調整する接続機構部分。パソコンでは、本体と周辺装置をつなぐ部分に、インターフェース・アダプタが使用されている。

インタプリタ　プログラムを 1 行ずつ読んで、内容を解釈しながら実行していくプログラム。（通常、ベーシックはROM（ロム）として提供される）

エム・エス・エックス（MSX）　ソフトウエアの、互換性のあるシステムのことで、ハードウエアにもソフトウエアにも使う。米国マイクロソフト社の提唱で、各コンピュータ・メーカーが開発。従来のパソコンは、各機種に独自のソフトウエアしか使用できなかったが、このシステムの開発により、MSX 仕様であれば、どこのメーカーのソフトウエアでも使用可能になった。

エラーメッセージ　パソコンを実行させたとき、命令の書き方や指定の方法、操作を誤ったりすると表示される文のこと。誤りの内容によって、いろんな種類のエラーメッセージがある。

エル・エス・アイ（LSI）　Large（ラージ） Scale（スケール） Integration（インテグレイション）の略で、大規模集積回路と呼ばれる。IC の一種で、トランジスタの素子（1000 個～10 万個）が、抵抗、ダイオードなどと回路配線で結ばれている。LSI が開発されたことによって、今日のパソコンのように、小型で高性能なコンピュータが開発された。

オーエス（OS）　Operating（オペレーティング） System（システム）の略。コンピュータを使いやすく、効率よく使うために用意された管理用のプログラムのこと。使用者のプログラムはすべて、OS の監視のもとで実行される。パソコンの代表的な OS として、CP/M、MS-DOS などがある。

オプション　パソコン本体に対して自由に選択ができ、追加すればいっそう便利になる周辺機器のこと。たとえばゲームに便利なジョイスティックとか、補助記憶装置のフロッピーディスク、カセットテープレコーダ、印刷に使うプリンタなど。

【カ行】

カーソル　パソコンのスイッチを入れると、ディスプレイ上に正方形のマーク、あるいは線分が点滅する。それをカーソルという。キーボードから入力すると、カーソルの位置に入力した文字、数字などが表示されるようになっていて、1つの文字あるいは数字などが表示されると、次の位置に移る。

カートリッジ　本来、詰めかえるという意味。ROM(ロム)カートリッジ、RAM(ラム)カートリッジを外部から接続し、ROM(ロム)の機能拡大、RAM(ラム)の記憶容量の拡大などに使う。

簡易言語　パッケージソフトのように完成されたプログラムではなく、使う側にプログラムの知識がなくても、いくつかの指令の与え方を覚えるだけで、簡単な計算、作表などの処理ができるようにしたもの。

関数　ベーシックの中には、あらかじめさまざまな関数が組みこまれている。関数を入力することによって計算できる。たとえばSIN（サイン）、COS（コサイン）、TAN（タンジェント）、LOG（ログ）、RND(ランダム)（乱数）など。

キーボード　パソコンの主要な入力装置。アルファベットや数字、記号などいろいろな種類のキー（けん盤）が並んでいる。キーを押すことによって、プログラムやデータなどを入力する。キーボードにはASCII(アスキー)キーボードとJIS(ジス)キーボードがあるが、日本ではカナ文字を使用するため、一般にはJIS(ジス)キーボードが使用されている。

機械語　コンピュータの中心部分であるCPU(シーピーユー)が直接理解できる言語。最終的には2進数で0と1の組み合わせになる。

行番号　ベーシックなどでプログラムを書く場合、プログラムの各行の頭に番号をつける、その番号のこと。ふつう、コンピュータは行番号の順番にプログラムを実行していく。

グラフィック　パソコンでは、ディスプレイに絵や図形が書ける機能を持った機種が多い。絵や図形のことを、グラフィックと呼ぶ。グラフィック画面は、ヨコ640×タテ400ドット（パソコンではもっとも多い）、ヨコ512×タテ256ドットなど、細かい点の集まりでできている。点を表示したり、点を連続して表示することによって線を描いたり、線の中の点を全部ぬりつぶしたりすることによって、絵や図形を描く。

グラフィック・キー　キーボード上で、GRPHとかGRAPHと書かれているキーのこと。このキーを押しながら文字キーを押すと、グラフィック文字が画面に表示される。グラフィック文字は、表示されていない機種もある。

クリック音　キーボードのキーをたたくと聞こえる、ピッピッという音のこと。

ケーブル　電線のたばを、電気絶縁体で外装したもの。パソコンと周辺機器を結ぶコードのことを、ケーブルと呼ぶことがある。

高水準言語　プログラミング言語の中で、フォートラン、コボル、パスカル、ベーシックなどを総称して、高水準言語という。機械語とかアセンブリ言語に比べて、日常語（といっても英語だが）に近いため、使いやすい。

コボル(COBOL)　COmmon(コモン) Business(ビジネス) Oriented(オリエンティッド) Language(ランゲージ)の略。1960年ごろ、米国国防総省の提案でつくられた一般事務処理用の高水準言語の一種。ステートメントが英語の文章に似ており、データ処理が得意。

コマンド　指令という意味。使用者が直接コンピュータを制御するために使う。たとえばベーシックでは、プログラムを入力したのち、RUN(ラン)と押せば、そのプログラムを実行するというぐあい。コマンドにはほかにNEW(ニュー)、LIST(リスト)、AUTO(オート)、CONT(コンティニュー)などがある。

コマンドレベル　ディスプレイ上にカーソル

が点滅し、Ok 表示が出ているときは、マイコンが入力を待っている状態で、この状態のことをコマンドレベルという。初期画面は、コマンドレベルである。プログラムを実行中は、コマンドレベルにはならない。

コントロール・キー　キーボード上でCTRLと書かれたキーのこと。このキーを押しながら文字キーを押すと、決められた特定の働きをする。たとえば、CTRL を押しながら S を押すと STOP と同じ働きをする。

コンパイラ　フォートラン、コボル、パスカルなどの高水準言語を、機械語になおすプログラムのこと。プログラム全部を機械語になおしたあとに実行に移る。

【サ行】

サウンド　パソコンにはシンセサイザー機能を持った機種があり、作曲を行なったり、演奏などができる。また、音声を合成してしゃべらせたりもできる。それらの機能の総称。

サブルーチン　1つのプログラムの中には、何度も同じ処理をくり返す部分や、特定の機能を持つ部分がある。それらの各部分をひとまとめにしたものをサブルーチンという。これを使うことによって、プログラムは短く簡単になる。

シーピーユー(CPU)　Central(セントラル) Processing(プロセッシング) Unit(ユニット)の略で、中央処理装置と呼ばれるコンピュータの中心部。パソコンの場合には、1個のLSIでできており、とくにマイクロプロセッサということもある。コンピュータはCPU、メモリ、I/Oで構成されているが、CPUはメモリから命令をとり出し、解釈し、実行する。

シフト・キー　キーボード上にある、SHIFTと書かれたキーのこと。キーには1つの文字だけでなく、記号など3つも4つも書かれたものがあり、シフト・キーと組み合わせることによって、これらの文字、記号などを使い分ける。

16進法　10進法では、1から9までかぞえて次が2ケタの10となるが、16進法だと、16番めが2ケタの10という表示になる。10進法の10から15に対する数字は、アルファベットのA～Fをあて、1ケタで数をあらわす。コンピュータは2進法だから、そのままあらわすと非常にケタ数が多くなるが、16進法では4ビット（4ケタ）をまとめて1ケタで表現できるので、データなどの表現に利用されている。

ジョイスティック　入力装置の一種。テレビゲームについているような操作用の棒のこと。前後左右など、任意に動かすことによって、ディスプレイ上の座標値を操作できる。

スクロール　ディスプレイ上に表示されるプログラム、データなどを上下、左右に動かすこと。

ステートメント　パソコンを働かせるためには、働かせる内容と手順を書いたプログラムが必要になる。プログラムは何行かの命令文の集まりになっていて、その1つひとつの命令文のことを、ステートメントという。

スペース・キー　キーボードの下のほうにある、長くて何も書いていないキーのこと。スペース・キーを押すと、1字ぶんのスペースが入力される。

ソフトウエア　ハードウエアに対比して使用される言葉。一般的には、プログラムのこと。もう少し広く、コンピュータの使用法や情報のこともソフトウエアということがある。

【タ行】

第5世代コンピュータ　コンピュータは、これまで回路素子によって第1世代・真空管、第2世代・トランジスタ、第3世代・IC、第4

世代・超LSI、というように発展をとげてきた。現在は第4世代・超LSIの時代。そこで、次代をになう未来型コンピュータを称して、第5世代コンピュータという。これまでのコンピュータと違い、連想、類推、判断、学習ができる「みずから思考するコンピュータ」になるといわれている。

ダイレクトモード　パソコンに指示を与えて何かを実行させる方法には、ダイレクトモードとプログラムモードの2種類がある。ダイレクトモードとは、直接コンピュータに命令を与える方法。たとえば、キーボードから「PRINT 2+3」と打って、RETURN（リターン）キーを押すと、すぐ実行されて、5と表示される。

ディスプレイ　CRTディスプレイともいう。テレビのブラウン管のことで、出力装置の代表的なもの。家庭用のテレビを、ディスプレイとして使用することも可能。ただ、パソコン専用のモニタテレビのほうが、映像が鮮明である。

データ　数値、文字、記号など。コンピュータはそれらのデータを入力し、計算、分類などの処理をほどこし、データを加工して出力する。このため、コンピュータの処理をEDP(Electronic（エレクトロニック） Data（データ） Processing（プロセッシィング）)と呼ぶこともある。

テープレコーダ　補助記憶装置の一種でプログラム、データなどを記録する。一般のオーディオ用カセットレコーダが利用できる。

デジタル　データを、数字などで表現すること。連続的に表現するアナログに対して、デジタル時計のように、一定の区切りごとに表現する。現在のコンピュータは、ほとんどがデジタル・コンピュータである。

デバッグ　プログラムの誤りを修正すること。プログラムの誤りのことをバグ（虫）といい、そのバグをとり出してなおす意味からデバッグという。

テン・キー　キーボードはJIS（ジス）によってアルファベットなどメインのキーが中央に配置されている。テン・キーとは、その右側にあり、0～9の数字が記されたキーのこと。数字の入力に便利である。機種によっては、テン・キーのないものもある。

ドット　点のこと。ディスプレイ画面はたくさんのドットで構成されていて、ヨコ640×タテ400ドットとか、ヨコ512×タテ256ドットなどと表記される。またプリンタには、ドット・インパクト方式のプリンタがあり、これは、紙にドットを打って印字する方法である。

【ナ行】

2進法　コンピュータは、電流が流れているか流れていないか、つまりON（オン）かOFF（オフ）かの状態しか理解できないため、ON（オン）に1、OFF（オフ）に0を対応させることによって、すべての数をあらわす。2の単位でケタ数（すう）が上がるため、2進法という。10進法の1、2、3、4は2進法では1、10、11、100となる。

ヌルストリング　文字変数に何も入っていない状態のこと。たとえば、「A$= 〃」と打つことによって、A$に何も入っていない。これが、A$がヌルストリングの状態である。DIM（ディメンション）文で文字変数を設定した場合には、文字変数はすべてヌルストリングになる。

【ハ行】

バージョン　改訂版のこと。改訂そのものを指すこともある。ふつう、最初のものをバージョン1とし、以後、改訂するごとにバージョン2、バージョン3……と版の番号をあげていく。

ハードウエア　本来、金物という意味。コン

ピュータ本体、周辺機器、装置など形のあるもののことを指す。ソフトウエアの対照語として使用される。

バイト　コンピュータの情報単位のことで、8ビットを1バイトという。1バイトで2の8乗すなわち256個の文字が区別でき、数字、アルファベット、カナなどを表現するのに用いられる。メモリ容量は、このバイト単位であらわすことが多い。

パソコン　Personal（パーソナル） Computer（コンピュータ）の略。小型で安価な卓上コンピュータのこと。個人でもコンピュータを買って使うことが可能になり、ホビーや計算、簡単な事務処理などに使われている。

パラメータ　ステートメントの内容を具体的に指定するための、変数の一種。たとえば「COLOR　3, 6, 11」と表示した場合の3、6、11の数字のこと。この例で、第1パラメータの3は表示色が明るい緑、第2パラメータの6は背景色が暗い赤、第3パラメータの11は周辺色が明るい黄色であることを指定している。

半導体　電子を流す導体と、流さない絶縁体の中間に位置するものを半導体という。ゲルマニウム、シリコンなどが代表的である。トランジスタは、半導体の発見によって開発された。現在は、半導体を利用してLSI、超LSIがつくられている。

ビット　コンピュータの情報量の最小単位のこと。2進数の0か1かの情報を1ビットという。8ビットとは2の8乗、256の情報量をあらわす。

ファームウエア　ハードウエアとソフトウエアの中間に位置しているものをいう。インタプリタもそのひとつ。

ファンクション・キー　コマンド、ステートメントなどの中で、よく使われるものを専用化してあるキーのこと。キーボード上でいちいち全部のアルファベットを打たなくても、ファンクション・キーを押すだけでその文字が表示される。

フォートラン(FORTRAN)　FORmula（フォーミュラ） TRANslator（トランスレイタ）の略。1954年ごろ、米国のIBM社で開発された高水準言語の一種。演算ステートメントがふつうの演算式によく似ているため、おもに科学技術の計算用に用いられる。

プリンタ　出力装置の一種。メモリの内容を印刷して打ち出す。パソコンのプリンタには、おもにドット・インパクト式と感熱式とがある。ドット・インパクト式は、ドット（点）を紙に打ちつけて印字するもの。感熱式は、特殊な用紙を必要とする。

フローチャート　アルゴリズムを、プログラムにしやすい形に整理して書いた論理的な処理手順をあらわす図。コンピュータに仕事をさせるためには、プログラムが必要だが、プログラムはすぐ書けるというものではないので、プログラムを書く準備としてフローチャートを作成する。

プログラム　コンピュータに仕事をさせるためには、仕事の内容と手順を指示する必要がある。その仕事の内容、手順をあらわしたものがプログラムである。ベーシックでは頭に行番号があり、そのあとにステートメント（命令文）が並んでいる。

プログラムモード　パソコンに仕事をさせる方法には、ダイレクトモードとプログラムモードがあるが、一般的にはプログラムモードで仕事をさせる。プログラムを入力し、そのプログラムの内容にそって行番号順に実行させる方法。実行させるときはRUN（ラン）を入力する。

プロセッサ　処理装置と訳される。コンピュータ・システム内では演算処理を行なう中枢

部を指す。

フロッピーディスク装置　パソコンの補助記憶装置の一種で、磁気カセットテープと同じ働きをする。磁気性の円盤（ディスケット）をセットして、情報の入出力を行なう装置。記憶容量が大きく、情報の入出力が磁気カセットテープより高速で行なえる、という特徴がある。

ベーシック(BASIC)　Beginner's(ビギナーズ) Allpurpose(オールパーパス) Symbolic(シンボリック) Instruction(インストラクション) Code(コード) の略。初歩的な高水準言語として、パソコンを使用するうえでは欠かせない。もともと、米国で大型コンピュータの端末装置用として多くの人が使用できることを目標に開発されたため、英語、数式などを使って、ほとんど日常語と同じ表現でプログラムを作成できる。

変数　数学の方程式における変数とは違い、データを記憶するメモリの場所という程度の意味。たとえばA=3は、Aという場所に3の値を入れるという意味になる。また、変数という名前のとおり、Aの値はイコールの右側の値によって変化する。ほかに、文字や記号を扱うための文字型変数というタイプもある。

ボー　電信、電話などの伝送速度を測定する単位のことで、1秒間に送れるビット数をいう。

【マ行】

マルチステートメント　ふつう、ステートメントは1行に1つしか入っていないが、同じ行の中に複数のステートメントをつなげたものをマルチステートメントという。その場合、各ステートメントはコロン（：）という記号でつながれる。

メモリ　パソコンの重要な機能のひとつ。メモリは、RAM(ラム)とROM(ロム)で構成される。RAM(ラム)は読み出し、書きこみ自在のメモリ。ROM(ロム)は読み出し専用メモリ。キーボードやカセットテープレコーダから入力したプログラムやデータは、メモリに記憶される。

モード　様式、方法のことで、パソコンを働かせるための演算方式とか操作方法など。

【ラ行】

ラム(RAM)　Random(ランダム) Access(アクセス) Memory(メモリ)の略。読み出し、書きこみが自由に行なえるメモリのこと。キーボードやカセットテープレコーダから入力したプログラムは、このRAMに記憶される。しかし、パソコンのスイッチを切ると、入力したプログラムは消えてしまうので注意が必要。

ループ　プログラムの中で、ある条件を満足するまで、何回もくり返し処理をさせること。ベーシックではFOR(フォー)～NEXT(ネクスト)という命令を使ってループを行なう。

ロム(ROM)　Read(リード) Only(オンリー) Memory(メモリ)の略で、読み出し専用メモリのこと。数個のICで構成されている。基本的なプログラムが、コンピュータ・メーカーによって最初から組みこまれているメモリなので、スイッチを切っても消えてしまうことはない。インタプリタは、この中に組みこまれている。

【ワ行】

ワープロ　ワードプロセッサの略で、文書処理システムのこと。文書の入力、記憶、編集、印字などの機能を持っている。

ふろく 2

和文さくいん

ふろく 3 英文さくいん

●監修者紹介

日本ナレッジインダストリ株式会社（略称：JKI）

資本金＝7500万円、従業員＝200人、社長＝西尾　出。

社名からもうかがえるように、同社は高度情報社会の担い手としてソフトウェア技術、情報通信技術、制御技術を駆使して産業システムや社会公共システムなどの研究開発に幅広い実績をあげている知識・情報産業会社である。

取り組む業務内容は実に多様であり、社会システムの調査、研究および企業の高度情報システムに関するコンサルティングには特に強い。またニューメディア関連システム、マイクロプロセッサ応用システムの開発、さらに医療システムや交通システムの研究開発、教育システムの研究開発、および関連付帯事業としての技術手引書・書籍の執筆・監修など幅広い活動にも積極的に取り組んでいる。こうした分野で仕事をしたい人々に対しては常時門戸を開いているというから、関心のある人はその門を叩いてみるとよい。

本社所在地　東京都品川区西五反田7－22－17　TOCビル

本書についてのお問い合わせは(ご質問内容を詳しくお書きください)、必ずハガキまたは手紙にて、弊社編集部あてにご郵送ください。

MSX2 パソコン実践ゼミ

監修者　日本ナレッジインダストリ株式会社　1985年12月10日発行

発行者　若松親光

発行所　株式会社 西東社（せいとうしゃ）
東京都千代田区神田錦町3―15
電話番号　東京03(291)5815代表
郵便番号　101
振替口座番号　東京8―6499

ISBN4-7916-0345-1

読者の皆さまへ

弊社の出版物は、全国有名小売書店の店頭に備えてあります。実物をごらんになって、お買い求めください。

書店に品切れの節は、店員にご下命になれば、弊社から取り寄せてお渡しいたします。

ぜひ、お近くの書店をご利用ください。

株式会社 西東社（せいとうしゃ）

東京都千代田区神田錦町3丁目15番地
電話　東京（03）291－5815(代表)

PC-8001mkII　日本ナレッジインダストリ㈱監修

PC-8801mkII　日本ナレッジインダストリ㈱編著

FM-NEW7　日本ナレッジインダストリ㈱監修

FM-77　日本ナレッジインダストリ㈱監修

MSXパソコン入門　日本ナレッジインダストリ㈱監修

MSXパソコン・ゼミ　日本ナレッジインダストリ㈱監修

MSXゲームプログラムの作り方　大山 昇著

MSXパソコンのABC　立花 恵著

カメラ入門　矢田博彦著

ビデオ撮影テクニックのABC　岡村征夫著

最新版 地方公務員(初級)　小林嬌一編著

最新版 危険物取扱者(乙種第4類)　小西勝也編著

最新版 調理師　佐藤淳子編著

最新版 栄養士　藤原真昭編著

最新版 2級ボイラー技士　谷口茂夫編著

就職試験シリーズ① 一般常識問題　現代就職試験研究センター編著

就職試験シリーズ② 読み書き問題　現代就職試験研究センター編著

就職試験シリーズ③ 適性試験問題　現代就職試験研究センター編著

就職試験シリーズ④ 面接試験問題　現代就職試験研究センター編著

書名	著者
スポーツ・シリーズ① ソフトボール	山本政親著
スポーツ・シリーズ② バドミントン	小島一平著
スポーツ・シリーズ③ ゴルフ	荒井進三著
スポーツ・シリーズ④ 野球	牧野茂著
スポーツ・シリーズ⑤ バレーボール	斉藤勝著
スポーツ・シリーズ⑥ バスケットボール	小浜元孝監修 竹市行男著
スポーツ・シリーズ⑦ 卓球	森武著
スポーツ・シリーズ⑧ 硬式テニス	渡辺力著
スポーツ・シリーズ⑨ 軟式テニス	武笠康雄著
スポーツ・シリーズ⑩ ウインドサーフィン	浅野雅明著 前野やすし写真
スポーツ・シリーズ⑪ 軟式テニスの基本	小板橋史彦著
スポーツ・シリーズ⑫ 硬式テニスの基本	渡辺功監修
スポーツ・シリーズ⑬ テーピング	成瀬臣彦著
スポーツ・シリーズ⑭ サッカー	成田十次郎著
テニスの腕が上がる本	渡辺康二著
図解早わかり ゴルフ入門	高氏一郎著
図解早わかり 野球入門	千葉茂著
図解早わかり 軟式テニス入門	高橋隆著
図解早わかり 硬式テニス入門	柳恵誌郎著
逆転発想のゴルフ	柴田敏郎著



西東社 定価1200円

ISBN4-7916-0345-1 C2055 ¥1200E